TECHNICAL REPORT
NATICK/TR-04/012

AD________________

RECONFIGURABLE AND ADAPTABLE MICRO-ROBOTS

by
R. L. Tummala
R. Mukherjee
N. Xi
D. Aslam
S. Mahadevan
and
J. Weng

Michigan State University
East Lansing, MI 48824

April 2004

Final Report
June 1998 – July 2002

Approved for public release; distribution is unlimited

Prepared for
U.S. Army Research, Development and Engineering Command
Natick Soldier Center
Natick, Massachusetts 01760-5020

DISCLAIMERS

The findings contained in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of trade names in this report does not constitute an official endorsement or approval of the use of such items.

DESTRUCTION NOTICE

For Classified Documents:

Follow the procedures in DoD 5200.22-M, Industrial Security Manual, Section II-19 or DoD 5200.1-R, Information Security Program Regulation, Chapter IX.

For Unclassified/Limited Distribution Documents:

Destroy by any method that prevents disclosure of contents or reconstruction of the document.

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE *(DD-MM-YYYY)* 27-04-2004	**2. REPORT TYPE** Final Report	**3. OATES COVEREO** *(From - To)* June 1998 - July 2002

4. TITLE ANO SUBTITLE RECONFIGURABLE AND ADAPTABLE MICRO-ROBOTS	**5a. CONTRACT NUMBER** C-DA'AN02-98-C-4025
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) R. L. Tummala, R. Mukherjee, N. Xi, D. Aslam, S. Mahadevan and J. Weng	**5d. PROJECT NUMBER**
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Michigan State University Electrical and Computer Engineering Mechanical Engineering Computer Science and Engineering East Lansing, MI 48824	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Sponsor: Defense Advanced Research Projects Agency (DARPA) Microsystems Technology Office (Elana Ethridge) 3701 North Fairfax Drive Arlington, VA 22203-1714	**10. SPONSOR/MONITOR'S ACRONYM(S)**
	11. SPONSOR/MONITOR'S REPORT NUMBER(S) NATICK/TR-04/012

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES
Monitor: US Army Research, Development and Engineering Command, Natick Soldier Center,
ATTN: AMSRD-NSC-SS-MA (T. Gilroy), Natick, MA 01760-5020

14. ABSTRACT

This report documents the mechanical structure, kinematic models and control implementation of two micro-robots, called Flipper and Crawler, that can traverse both on walls and floors. They are battery powered and use specially designed feet using suction cups. The design of these feet is described in Section 2. The mechanical design was arrived at through extensive analysis. It has a biped structure with revolute and prismatic hip joints. The underactuated mechanical structure was chosen to reduce weight and power consumption by decreasing the number of actuators. Micro-controllers from Motorola, Inc. and Texas Instruments, Inc. were used as embedded controllers with the ability to implement motion planning and trajectory generation algorithms to control these robots. The details are given in Sections 3 and 4. The report concludes with the summary and suggested improvements.

15. SUBJECT TERMS

RECONFIGURABILITY	MECHANICAL STRUCTURES	BATTERY POWERED	ACTUATORS
ADAPTABILITY	KINEMATIC MODELS	SUCTION CUPS	ROBOTICS
MINIATURE	POWER CONSUMPTION REDUCTION	MICRO-ROBOT	
ROBOTS	EMBEDDED CONTROLLERS	MICROBOTICS	

16. SECURITY CLASSIFICATIDN OF:			**17. LIMITATION OF ABSTRACT**	**1B. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Thomas Gilroy
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAR	57	**19b. TELEPHONE NUMBER** *(Include area code)* 508-233-5855

Contents

Contents (Cont'd)

List of Figures

List of Tables

PREFACE

This report outlines the research undertaken by Michigan State University, East Lansing, MI, to develop the mechanical structure, the kinematics model, and the control implementation of two wall-climbing micro-robots. These robots have the capability to walk on both horizontal and vertical surfaces, and transit between some inclined surfaces. The purpose of this report is to communicate the design, implementation and evaluation of these unique robots. Detailed listing of software, hardware and mechanical drawings will be available as a separate document. The project was completed during the period June 1998 to July 2002 under contract number C-DAAN02-98-C-4025, under the direction of U.S. Army Research, Development and Engineering Command, Natick Soldier Center, Natick, MA, and sponsorship of the Defense Advanced Research Projects Agency (DARPA).

RECONFIGURABLE AND ADAPTABLE MICRO-ROBOTS

1. Introduction

The multidisciplinary miniature robotics design team at Michigan State University has developed prototype miniature climbing robots that can be used as remote sensors for gathering information about a hostile situation in a building, fire and rescue operations and security. They are capable of traveling along horizontal and vertical surfaces such as floors, walls and ceilings. They can adapt to different smooth surfaces such as glass, steel and drywall or painted walls. They are semi-autonomous and capable of carrying wireless sensors, such as camera or microphone and their transmitters. These robots are designed to be lightweight to minimize power consumption and sufficiently small in size so that they can travel through confined spaces. They are named **Flipper** and **Crawler** for discussion purposes are shown in Figure 1 and Figure 2.

The purpose of this report is to communicate the design, implementation and evaluation of these unique robots. Detailed listing of software, hardware and mechanical drawings will be available as a separate document.

1.1 Background

Numerous large robots exist for climbing inclined surfaces. Most climbing robots use legged structures with two (biped) to eight legs, where more limbs inherently provide redundant support during walking and can increase load capacity and safety. However, these benefits are offset by complexity, size and weight. Thus, in situations where compactness and efficiency are critical, a structure with minimal weight and complexity is needed. For these reasons, we have chosen the biped format.

Most biped robots use similar ankle structures where articulation is provided to both feet and steering to at least one foot. Bipeds vary mostly in the style of middle joints. Robots using revolute middle joint include the robot by Nishi [1] and the robot ROBIN [2]. A prismatic middle joint is used by ROSTAMIV [3], and the robot by Yano [4] does not have a middle joint, but simply a rigid body.

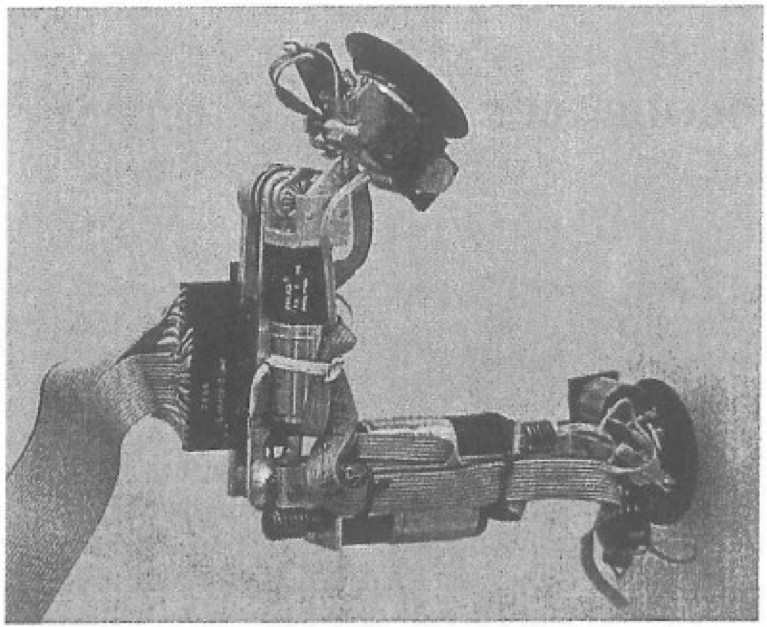


Figure 1. The Biped Climbing Robot (Flipper).

Beyond similarity in joint structure, our robots possess several characteristics not found in the aforementioned robots. First, these are the smallest of the bipeds found in the literature. For example, in its longest configuration, Flipper measures 45 mm × 45 mm × 200 mm and weighs approximately 335 grams. The crawler has a sectional area of 50mm ×80mm and weighs approximately 450 grams. The next larger biped is Yano's robot [4], which measures 380mm × 250mm × 170mm and weighs 10 kg.

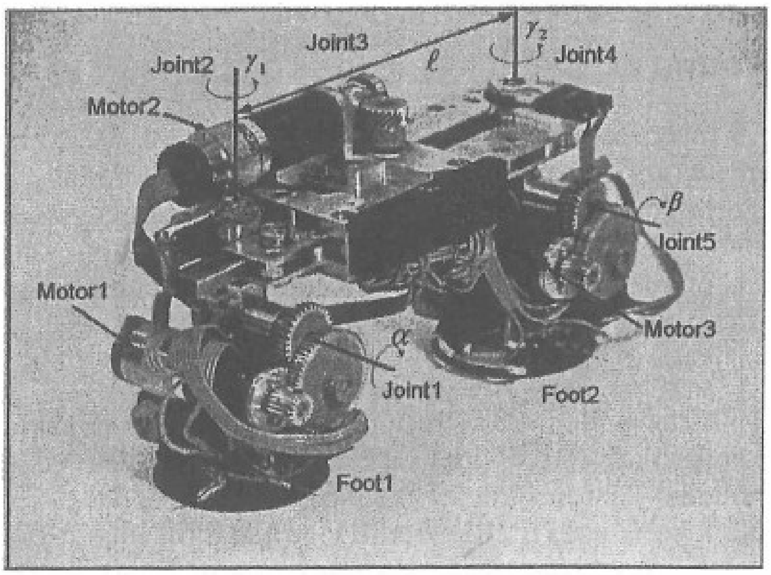


Figure 2. Biped Climbing Robot (Crawler)

A substantial feature contributing to the small size and weight for these robots are their under-actuated design. These under-actuated designs incorporate fewer actuators, but do not sacrifice mobility or other functionality. Such designs reduce the weight of the robot for two good reasons. First, the weight of each actuator constitutes a substantial portion of the weight, and second, elimination of the actuator allows us to downsize the remaining actuators since each actuator is designed to carry the weight of the others. A small penalty paid for under actuation is increased complexity of motion planning and joint control. There was sufficient computer power on board to handle this.

The report is divided into three sections. Section 2 describes specially designed robotic foot, called Smart Robotic Foot (SRF) using suction cups. A description of mechanical design, controller architecture and motion planning for Flipper robot is given in Section 3. Section 4 gives the details of the Crawler robot.

2. Smart Robotic Foot (SRF) Design and Fabrication

Climbing robots that have been developed in the past use either suction cups or magnetic grippers for attachment to vertical surfaces [5], [6], [7], [8]. Other novel methods such as propeller drive have also been designed. The suction cups used in the past have been quite large in footprint, size and weight. They are not self-contained as both power and air are supplied externally through a tether. This is not suitable in our case. So we have designed a self-contained suction cup based robot feet for our robots. We call these Smart Robotic Feet (SRF) [9] (Dangi and Aslam, 2000).

The suction cup based foot designed for our robots is shown in Figure 3. It uses a diaphragm type vacuum pump as it provides large pumping speeds for small suction volumes. The pump (model KNFNMP02 L/U) measures only 27.1mm × 16.9mm × 28mm in size and weighs 20 grams. It supplies a vacuum level of 10.34 Kpa and operates at 6V and draws approximately 48mA current. The pump is connected to a flat suction cup using a custom designed miniature aluminum connector. The connector also acts as mounting platform for the robot body and rest of the components. The pressure inside the suction cup is monitored using fully integrated signal conditioned pressure sensor mounted on the connector. A signal from the pressure sensor is used by the main control unit to decide whether the foot is firmly attached to the surface. A specially designed touch sensors are installed on the cup to provide tactile feedback. They are used to eliminate the steady state errors caused by the gravity and other errors in the control systems. A MEMS based microvalve is used release the cup. The fully assembled foot weighs approximately 35 grams with 40mm cup.

2.1 SRF Testing

In order to determine the upper limit on the weight of the robot body supported by the SRF, we conducted some experiments by using a mock-up of our micro-robot. Using this mockup, the weight supported by the SRF on different surfaces using parallel and perpendicular configurations was determined. In parallel configuration, the load is applied parallel to the surface at a distance *D* from surface. Each test was conducted on different surfaces. The surface conditions were also alternated between wet and dry. The wet condition was simulated by spray of de-ionized water from a distance of 15cm. These results are given in [9]. The body weight of the robot is selected based on these results. Figure 3 is the close-up of the robot foot with touch sensors.

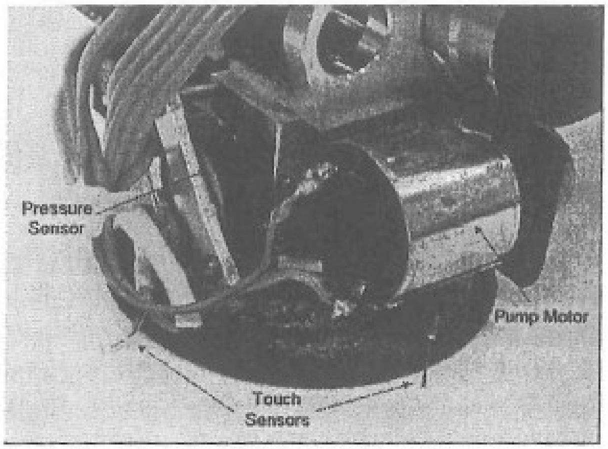


Figure 3. Robot Foot with Touch Sensors

3. Miniature Robot With Revolute Hip (Flipper)

3.1 Mechanical Design

3.1.1 Biped Structure

As described in the previous sections, the current format of the flipper robot consists of a two-legged structure using a revolute hip joint to support two legs of equal length. At the end of each of these legs are ankle joints supporting suction cup feet. Our robot has five links, Links 1 through 5, and four joints as shown in Figure 4. Link 1 or Link 5 is securely fixed to the traveling surface during articulation of the structure. The remaining four links are driven by three actuators, with one driving two links. Specifically, one actuator drives Joint 1 to steer the robot, a second actuator drives both Joints 2 and 3, and a third actuator drives Joint 4. Together, the second and third actuators drive Joints 2,3 and 4 to generate a cartwheel gait that involves flipping of the robot structure. The reduction of the number of actuators is accomplished by coupling articulation of joints 2 and 3 by a belt drive. The coupling is designed to maintain the rotation of Joint 3 equal to twice the rotation of Joint 2. However, the reduction of actuators accomplished this way also reduces the degree of freedom and complicates the motion planning task. This problem was overcome within the software. In Figure 4, α, β, λ, and γ denote joint angles.

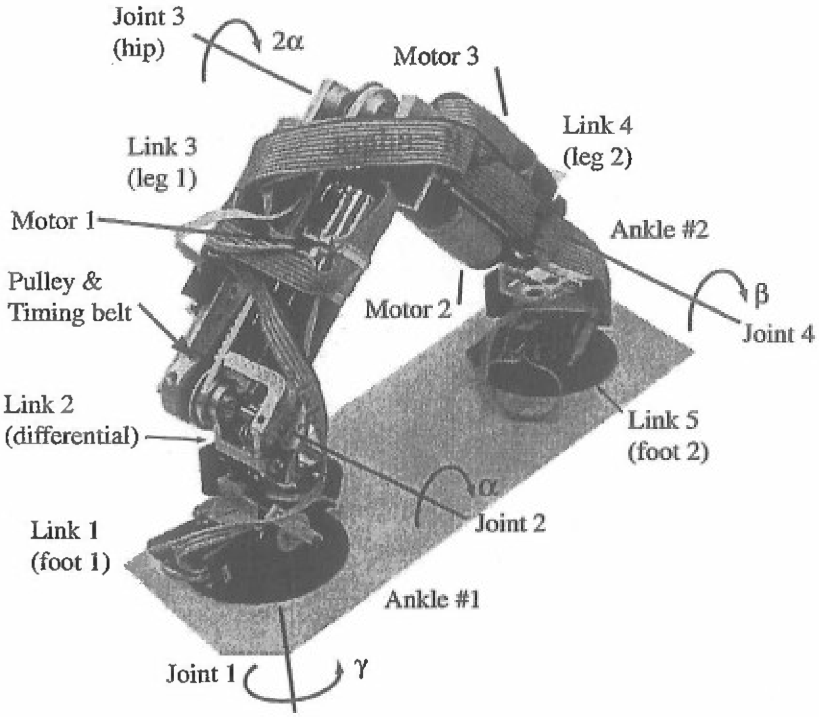


Figure 4. Miniature Robot with Revolute Hip Joint (Flipper)

3.1.2 Force Analysis

The purpose of performing this force analysis is to determine the size and power of the actuators. A static force analysis is performed on the robot in the configurations that apply the largest forces on the actuators. The robot was analyzed while climbing a vertical surface in a straight, cantilevered orientation where the maximum moments and loads are encountered. A safety factor of at least 2.0 was considered at this operating point. Such a safety factor should be sufficient to accommodate dynamic forces as well as unexpected friction. For example, some of the forces and torques on the robot supported by Foot 1, are shown in Figure 5, where:

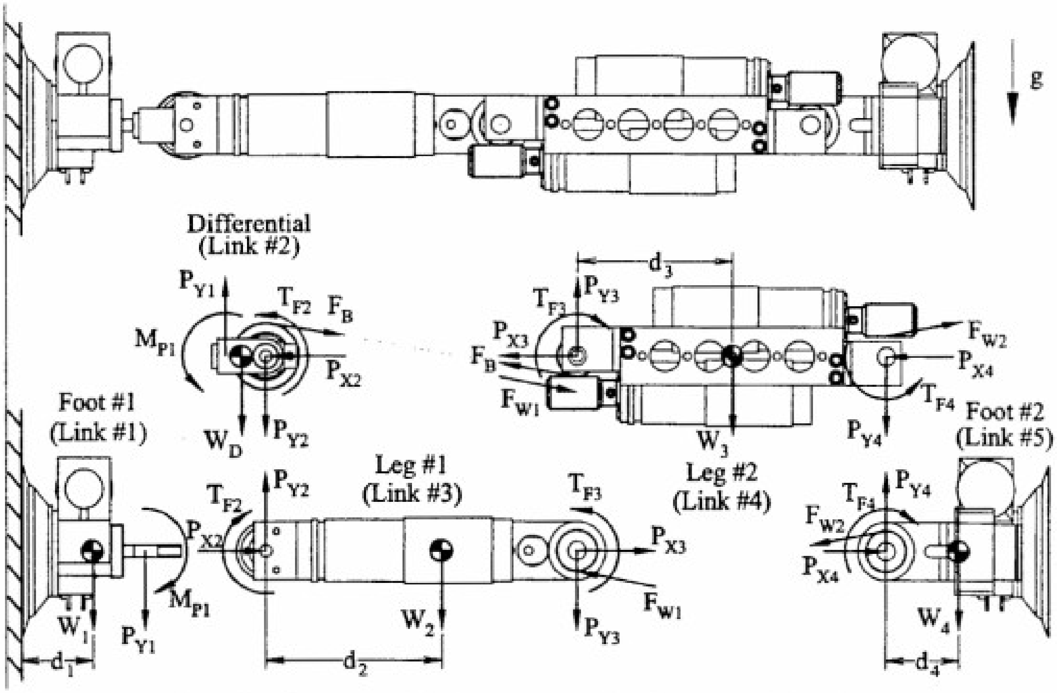


Figure 5. Free Body Diagrams of the Climbing Robot Supported by Foot #1 on a Vertical Surface (some forces omitted for clarity).

P_{Xi} = force on the i^{th} joint in the X-direction,
P_{Yi} = force on the i^{th} joint in the y- direction,
F_B = Belt force acting on the pulley,
F_{WT} = Tangential worm load,
F_{WR} = Radial worm load,
T_F = Static friction torques,
W = Weight of the body, and
d_i = Distance to the center of mass of the ith body from its support point.

The other parameters shown in the Figure arc:

r_g = Pitch radius of worm gear
r_{Si} = Radius of shaft supporting the i^{th} joint
α = angle of belt between pulleys
λ = worm lead angle
ψ = worm gear pressure angle
μ = Static coefficient of friction

Similar analysis was performed for climbing robot supported by Foot #2 and steering on Foot #2. For complete analysis, the reader is referred to [10].

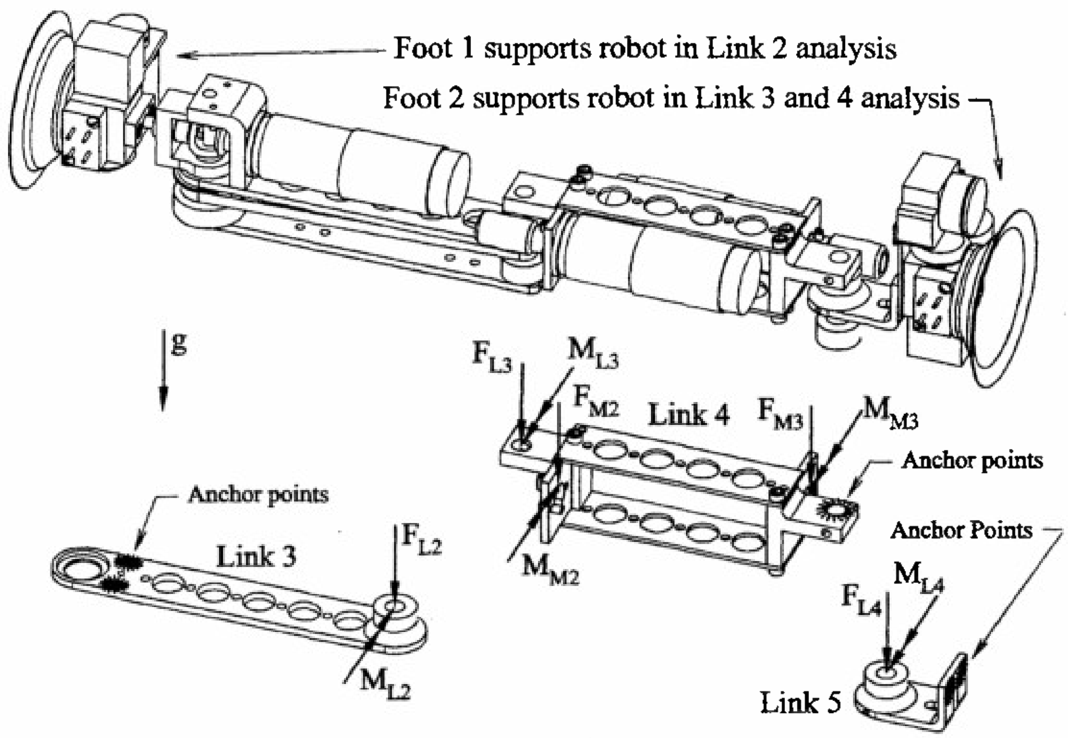


Figure 6. Diagram of Robot Boundary Conditions Used for Structural Analysis

3.1.3 Actuator Selection

The actuators considered were geared DC motors with position encoders. The size of motors considered range from 10mm diameter to 17 mm diameter, and 0.75 watts to 3.2 watts, respectively. Motor manufacturers with a broad product range, including motor selection, gear heads, and encoders, were the primary concern. These include API Portescap [11], Maxon Precision Motors Inc [12], and Micro Mo Electronics [13]. After considerable analysis of the needs, API Portescap motors were selected [10].

3.1.4 Structural Optimization

The next step in our design involved the examination of the structure of the robot to assure that it can support the given loads and to determine if its weight can be further minimized. FEM analysis of the structurc was performed to assure adequate strength and provide further weight reductions. The holes shown in Figure 6 on the structure are actually a product of these efforts. Complete analysis is given in [10].

3.2 Kinematics Model

Because the structure of robot requires that at least one foot remains in contact with the surface at all times, the coordinate frames were set up in three-dimensional space and divided into right-foot supporting (RFS) phase and left-foot supporting (LFS) phase. In the RFS phase, the right foot is chosen as the base coordinate frame. Similarly in the LFS phase, the left foot is chosen as the base coordinate frame. During a robot walk, the two phases (RFS and LFS) will be used one at a time, depending on which foot is anchored to the surface to support the robot.

The kinematics model of our robot design was described in [14] using the following arm transformation:

$$T = \begin{pmatrix} R & p \\ 0 & 1 \end{pmatrix} \tag{1}$$

where in the RFS phase the rotation matrix R and position vector p are:

$$R_{RFS} = \begin{bmatrix} C(\gamma+\alpha)C(\beta+3\alpha) & S(\gamma+\alpha) & C(\gamma+\alpha)S(\beta+3\alpha) \\ S(\gamma+\alpha)C(\beta+3\alpha) & -C(\gamma+\alpha) & -S(\gamma+\alpha)S(\beta+3\alpha) \\ -S(\beta+3\alpha) & -C(\beta+3\alpha) & 0 \end{bmatrix} \tag{2}$$

$$p_{RFS} = \begin{bmatrix} C(\gamma+\alpha)(-L_2C(2\alpha)+L_2C(3\alpha)) \\ S(\gamma+\alpha)(-L_2C(\alpha)+L_2C(3\alpha)) \\ L_1+L_2S(\alpha)-L_2S(3\alpha) \end{bmatrix} \tag{3}$$

in the LFS phase the rotation matrix R and position vector p are:

$$R_{LFS} = \begin{bmatrix} C(\gamma+\alpha)C(\beta+3\alpha) & -C(\beta+3\alpha)S(\gamma+\alpha) & S(\beta+3\alpha) \\ C(\gamma+\alpha)S(\beta+3\alpha) & -S(\beta+3\alpha)S(\gamma+\alpha) & C(\beta+3\alpha) \\ -S(\gamma+\alpha) & -C(\gamma+\alpha) & 0 \end{bmatrix} \tag{4}$$

$$P_{LFS} = \begin{bmatrix} -L_2C(\beta)+L_2C(\beta+2\alpha)-L_1S(\beta+3\alpha) \\ -L_2S(\beta)+L_2S(\beta+2\alpha)-L_1C(\beta+3\alpha) \\ 0 \end{bmatrix} \tag{5}$$

where $C(\cdot)$ and $S(\cdot)$ represent the cosine and sine functions, respectively, γ is the *controlled* rotation of Joint 1, and α is the *induced* rotation of the Joint 1 by Joint 3..

3.2.1 Controlling Single-Plane Walk

When the robot is in its RFS phase, the XZ plane of the supporting foot defines the robot's sagittal plane. In the RFS phase, the robot is walking/climbing in the X-Y plane. In order to keep the robot to walk straight along the X axis, one of the following conditions must hold:

1. $\mathbf{s} = [0\ -1\ \ 0]$. This condition constrains the robot to flip forward/backward in the X-Z plane defined by the coordinate frame of the supporting foot.
2. $\mathbf{p}_y = 0$. This condition constrains the end point to land at a location on the x-axis defined by the coordinate frame of the supporting foot.

It is obvious from Equation (1) and Equation (2) that both of these conditions can be satisfied when $\sin(\gamma+\alpha) = 0$. This implies that $\gamma+\alpha = 0$. Since α is the induced rotation, it behaves like a bound variable, the only free variable which can satisfy this constrain is γ, which must be set to $-\alpha$. In the LFS phase, $\boldsymbol{p}_z = 0$ in Equation (4) ensures that the main axis of the robot body will always move in the sagittal plane (the X-Y plane defined by the coordinate frame of the supporting foot). This also guarantees that the end point will land at a location on the x-axis defined by the coordinate frame of the supporting foot. However, to prepare for the next walking step motion, it is desirable to set the orientation of the end point properly. One way to achieve this is to set $\mathbf{s} = [0\ \ 0\ \ -1]$. It is also obvious from Equation (3), that this condition can be satisfied when $\sin(\gamma+\alpha) = 0$ and $\cos(\gamma+\alpha) = 1$ which is true when $\gamma + \alpha = 0$.

From these two observations of single-plane walk, it is clear that compensating the induced rotation α by a controlled rotation of $-\alpha$ on the steering foot is necessary to control the robot to move in a straight line.

3.2.2 Controlling Dual-Plane Walk

To control the robot to walk across two different planes, the end effector has to be aligned so it is perpendicular to the target plane, that is the **a** column of the arm transformation matrix has to satisfy $\mathbf{a} = k\mathbf{N}$, where **N** is the normal vector of the target plane.

In the LFS phase, the z component of the **a** vector in the rotation matrix is 0, so it is impossible to satisfy all values of N. Therefore, the robot cannot walk across two different planes when it is in its LFS phase. In the RFS phase, the **a** vector depends on three parameters: α, β, and γ, Therefore, in the RFS, the robot has one extra DOF to match its end effector normal to that of the target plane at *any* orientation.

3.3 Controller Hardware Design

The implementation of the controller is shown in Figure 7. We have chosen Motorola MPC555 for controlling the robot. It is a 32-bit Power PC based RISC processor and offers the following features.

□ 26KB fast RAM
□ 6KB TPU microcode RAM
□ 448 KB flash EEPROM with 5-Volt programming
□ Serial Interfaces, consisting of:
- Queued Serial Multi-Channel Module (QSMCM)
- Dual CAN 2.0B controller (TouCAN)

□ 50-channel timer system: dual time processor unit (TPU3) with 16 timer channels each, and
□ Modular I/O subsystem (MIOS1) with 18 timer channels.
□ Queued Analog Digital Convertcr (QADC)
□ Dual voltage supply (3.3 and 5 Volts)
□ 5-Volt I/O system
□ 40MHz operation speed

The hardware structure of the microcontroller is quite suitable for our application. It has the capabilities to receive signals from pressure sensors, touch sensors, and position encoders. It provides appropriate signals for the joint motors, suction cup pump motors and Shape Memory Alloy based microvalves. For example, the Multi lnput/Output Subsystem (MIOS) of the MPC555 is used for driving circuits that require digital switching, such as micro-valves and motor pumps. The MIOS also handles various digital input and output signals (touch sensors, receiver data lines). For each joint motor, MIOS will output PWM and direction signals. Quadrature output from each motor encoder is processed by the TPU via the Fast Quadrature Decode function. Analog output from pressure sensor and potentiometer are connected to the QADC analog input channcls. Output from the receiver (Rx) can be connected either to QADC or MIOS.

The total size of the executable code for controlling Flipper is about 150K bytes. The 448K flash EEPROM is large enough to hold thc code.

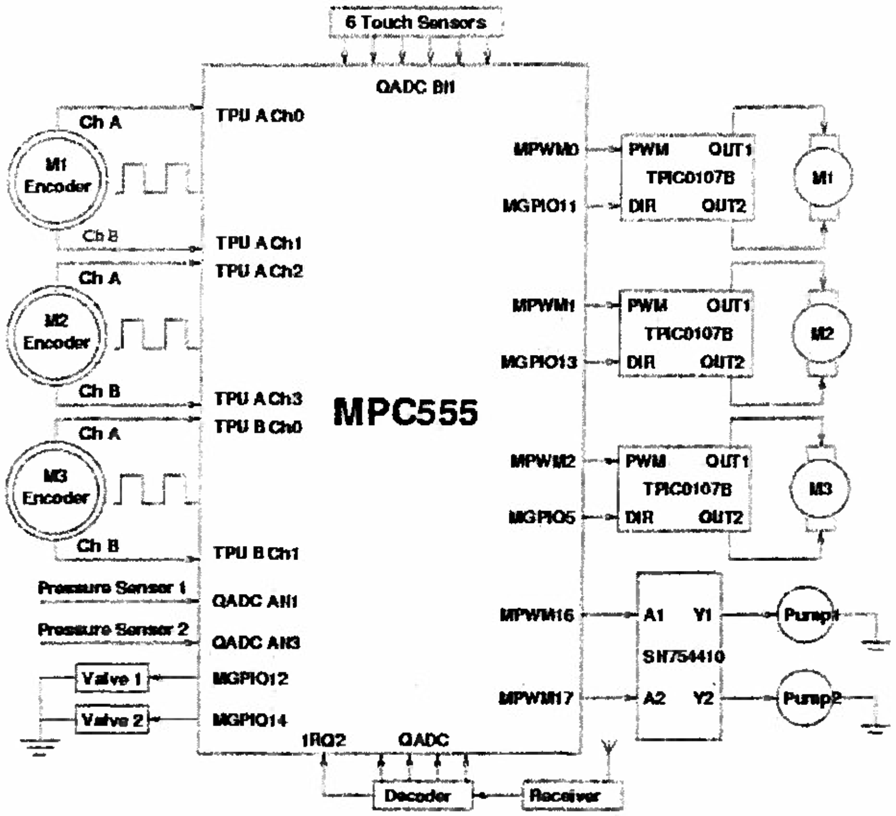


Figure 7. MPC555-Based Robot Controller

3.3.1 Motion Control System

The architecture of the motion control system is shown in Figure 8. The system has been designed to enable the Flipper to accept high level composite commands such as "Move Forward", "Move Backward", "Turn", "Climb", etc. These high-level user commands are considered composite because each requires simultaneous control of more than one joint and actuator and processing of input signal from several sensors (touch sensors, pressure sensors, etc).

The Robot Commander is a subsystem that directly accepts user input commands from a remote transmitter or from a serial communication line. The `select()` function implemented in the C library routine emulates the Unix `select()` system call that enables I/O multiplexing from different sources. Internally, the `select()` system call blocks until an interrupt from the receiver or serial line is received.

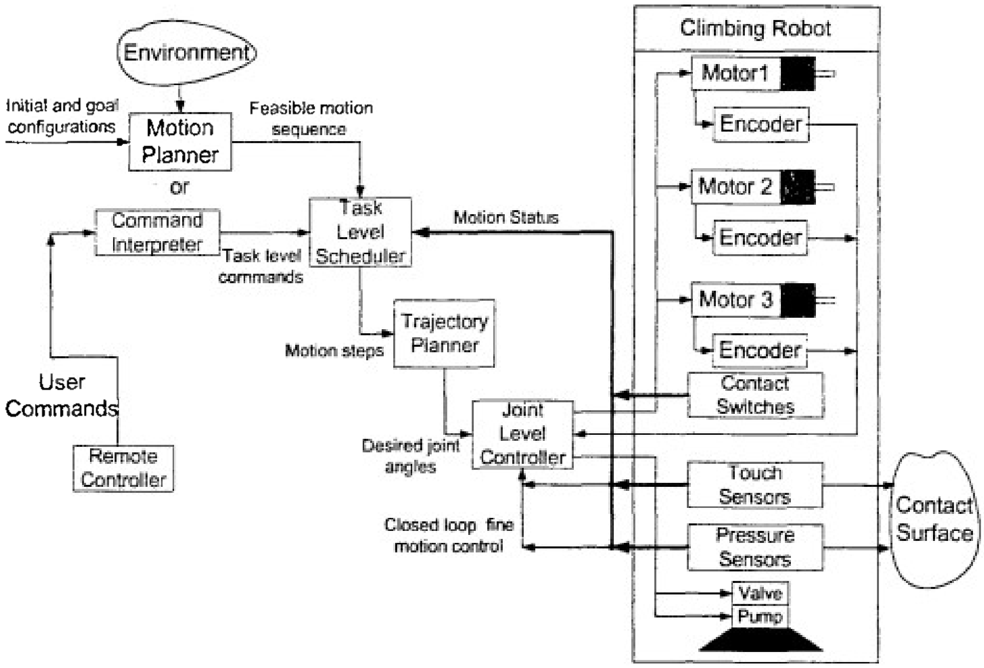


Figure 8. Controller Hierarchy

The Command Interpreter, a Finite State Machine (FSM) is used to check the safety and validity of a given command. The FSM also enables thc robot to be controlled by remote transmitter with a small number of button commands. A safe and valid command is decomposed into a combination of simpler commands to be processed by the Command Interpreter. For instance, when the Flipper is commanded to "Move Forward" it will continuously perform an alternating sequence of "Flip1" and "Flip2" commands. The two "Flip" commands are necessary because the two ankle joints of the Flipper have different structure. The sequence of operations for performing a "Flip" when the robot is being supported by Foot 1 is different from the sequence when the robot is standing on Foot 2.

When the Command Interpreter is executing a command, possibly more than one joint and sensor have to be monitored simultaneously. For each command it recognizes, the Command Interpreter knows which joints and sensors are used and whether these joints **can** be controlled sequentially or **must** be controlled in parallel. Access to individual joints and sensors is handled directly by the Joint Level Controller subsystem.

3.3.2 Remote Control System

To enable remote operation, we have built a transmitter and receiver for controlling our robot. When designing this system, we had two choices: RF (radio frequency) or infrared signal. Thc availability of remote control for home appliances (TV, VCR, audio system, etc.) enticed us to use infrared signal but their use is limited by direct line of sight between the remote and the robot. For this reason we built the remote control unit around an RF transmitter manufactured by Glolab (http://www.glolab.com). The picture of the

16-key remote control unit is shown in Figure 9. The key sequences encode a set of operator commands.

Incoming RF signals from the remote control unit are processed by MPC 555. To enable reception of asynchronous data from the remote control unit, interrupt driven software was incorporated.

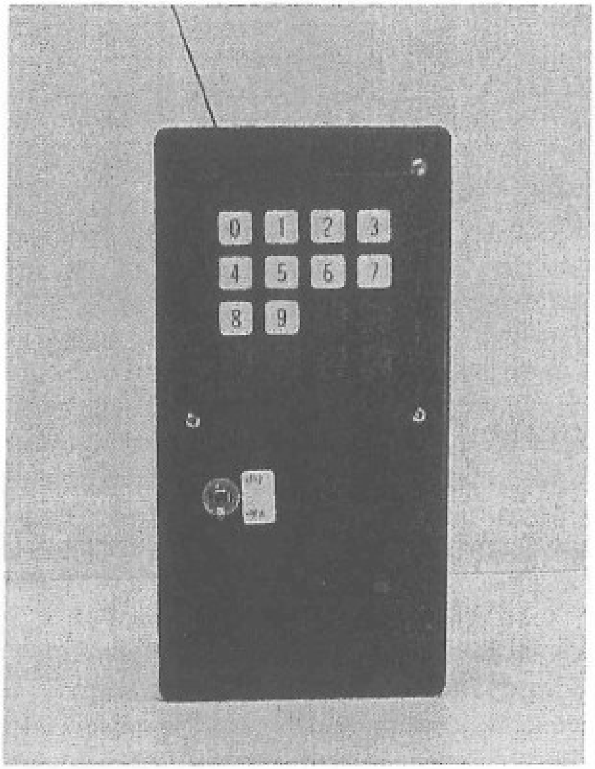


Figure 9. Remote control unit

3.4 Controller Software Development

3.4.1 Software Modules

To provide software interface to various units in the MPC555 processor, several C and C++ routines have been written. These routines are needed to implement low-level functionalities such as: interface to the MIOS (digital I/O, PWM generation), Analog/Digital Converter, Serial Communication, Time Processing Unit, Flash Memory Management, and Unified System Interface Unit. Most of these low-level routines were implemented in C and PowerPC assembly. A detailed discussion is available in the software manual [15].

Higher-level functionalities were built on top of these low-level modules. Some examples of the higher level functionalities are: C stdio-like routines, memory allocation routines, boot reset routine, command interpreter, PID (Joint) controller routines, kinematics calculation routines, etc. Most of these higher-level routines were implemented in C++ and classes such as `Analog`, `Clock`, `CommandInt`, `Digital`, `Joint`, `MPC`, `PIDControl`, `Suction`, `Task`, `Scheduler` were defined.

The following sections describe important software components implemented in our controller. A more comprehensive and elaborate description of the software is given in the separate Programmer's Manual document.

3.4.2 Command Interpreter

The receiver on the controller board allows Flipper to be controlled from a 16-key remote transmitter. To enable a very large number of selections sent from the 16-key transmitter, the command interpreter employs a finite state machine (FSM) in decoding the key sequence. With a k-state FSM the number of possible selections is $16k$.

In the current implementation, the set of operator commands can bc catcgorized into two different types: motion commands and actuation commands. High-level motion commands include ``Move Forward", ``Climb", ``Move Backward", ``Turn x degrees", etc. Actuation commands include ``Turn Suction On", ``Turn Valve On", etc.

The purpose of employing the FSM is twofold: *user-friendliness* and *safety*. As described in Section 3.2, the Flipper defines two phases, RFS and LFS. A ``Move Forward" command, for instance, should activate different kinematics control modes depending of the robot's current phase. Using a remote transmitter enables a human operator to send commands in any order. However, there are sequences of unsafe commands for the robot. For instance, if the robot is being supported on only one of its feet, the command for turning off the pump of that foot should be prohibited. To provide safety features to thc robot, the command interpreter should reject this request based on the current state of the FSM.

In software, the Command Interpreter is implemented as two C++ classes: `Module` and `CommandInt`. The class `Module` is an abstraction of dynamically callable function for handling a certain command, not a module in the MPC555 architecture. When a `Module` object is created, the following parameters are passed to the constructor:

□ Pointer to the dynamically callablc function for the module,
□ A set of initial states where the function can be called from,
□ The final state to be set in the FSM when the function finishes execution, and
□ A command key associated with this module

Each dynamically callable module must be registered to the `CommandInt` object by calling the `AddModule()` method. This method will then add the pointer to the callable function to an associative map when the command key is used for the search key in the map.

3.4.3 Serial Communication Module

To provide interface to the Queued Serial Module (QSM) on the MPC555 micro controller, the `QSM_map` structure is defined in the header file `qsm.h`. This module provides routines for initializing and resetting the MPC555 QSM, checking available input data, sending and receiving bytes to the serial communication line.

The serial communication module provides basic character-based I/O needed by the stdio replacement module (more details in the Programmer's Manual). Standard I/O operations for performing input/output to/from *stdin, stdout*, and *stderr* are defined to have a FILE parameter, similar to those in Unix system calls. However, since we are working in an embedded system, a file system does not really exist. In our implementation, a "FILE" is a logical source of I/O operation, which can be one of the following sources:

□ Serial communication line,
□ Glolab Receiver.

3.4.4 Servo Control Module

To encapsulate all the processing detail of a PID controller, the `PIDControl` class was created. Partial listing of the interface to this class is given in the following listing.

```
class PIDControl {
   public:
      SetMoveDist (int dist);
      void ServoMove();
};
```

The function PIDControl::ServoMove() is crucial in the implementation of the PID control algorithm. When invoked, this function will first get the current value of the motor encoder. The Fast Quadrature Decode (FQD) function in the TPU used for this purpose returns a signed 16-bit value. However, the PIDControl class stores all the position-related variables as 32-bit values. Hence, in order to enable the use of a larger range of position information, some arithmetic manipulations have to be carried out. Here, two cases have to be considered. They are: 1) when the FQD counter is counting up and reaches a maximum value of 0x7FFF (32767) and 2) when the FQD counter is counting down and reaches a minimum value of 0xFFFF (-32768). In the first case, overflow occurs and in the second, underflow occurs. On overflow, the encoder value will "plummet" from 32767 (the largest signed 16-bit integer value) to -32768 (thc smallest signed 16-bit integer value).

Both the overflow and underflow can be detected in softwarc by considering the physical limit of the motor and the position encoder. Our robot uses the Escap 17N78 motors with winding types 216E and 210E. Among the two, the 210E winding has a higher speed, which is 8300 rpm.

With a servo updatc rate of 1KHz (once every millisecond), the maximum speed of this motor would be approximately 9 encoder counts per millisecond. Therefore, an unusually big "jump" in the detected spced indicates the occurrence of underflow or overflow. The correction is made by adding or subtracting a hexadecimal value of 0x10000 (32768 in decimal).

To have a smooth operation, each joint motor is controlled using a trapezoidal velocity profile. Initially the motor has a zero velocity. To reach a destination position, the motor is first accelerated up to a maximum velocity limit. Upon reaching this limit, the motor will run at a constant velocity for a period of time. Finally, the motor will be decelerated back to zero velocity to reach the destination. The shape of the profile is determined by the acceleration and velocity limits and the actual distance to be traveled by the motor. At each servo tick, the trapezoidal profile determines the *desired velocity* of the motor. However, by integrating the velocity over time, the desired position of the motor can be obtained indirectly. The PID controller will take the desired position as the reference signal. It then computes the error e(t) from the difference between the desired position and the actual position of the motor as decoded by the Fast Quadrature Decode function in the TPU. Once the error is calculated, the output u(t) is determined and the necessary output signals, i.e. the magnitude (PWM signal) and direction (digital output), are generated.

In order to have a finer control over the trapezoidal profile, all position, velocity, and acceleration variables are considered as having an implied 8-bit fraction. Therefore, the 32-bit data are viewed as 24-bit integer part and 8-bit fraction part. All calculations involving position and velocity values are carried out as fixed-point arithmetic operations with an implied decimal point after the 8th least significant bit. However, the final PID output is obtained only from thc integer part.

3.4.5 Controlling Foot Orientation

The suction feet will stick to a surface when it is oriented at a normal angle to the surface. Misalignment of the feet will prevent them from making a firm grip on the surface.

Therefore, positioning the feet at a normal position to the surface is an important subtask of our robot.

The foot orientation is represented by the **a** vector in Equation (2) and Equation (4). The ideal foot orientations correspond to **a** = [0 0 –1] in the RFS and **a** = [0 1 0] in the LFS phase. Both of these vectors are measured on the coordinate frame of the supporting foot. Ideal foot angle can be obtained by solving for the joint parameters α, β, and γ. Since the errors caused by gravitational forces at various positions of the robot cannot be accurately determined, touch sensors are installed on the foot to obtain feedback on the foot orientation. These sensors are shown in Figure 3. This feedback also helps to speed up the operation of foot placement.

3.4.6 Interrupt Handler

One of the exception vector offset provided by the PowerPC architecture is the External Interrupt Exception. In this experiment, external interrupt is used for many different purposes:

□ Periodic Interrupt Timer handler: used for activating the PID Servo controller and multi-tasking facility,
□ Time Base Interrupt: used for alarm-like feature
□ Decrementer Interrupt: used for timer count down facility
□ Serial Communication Interrupt
□ Receiver Data Ready
□ Key-pad Data Ready

The PowerPC 555 processor provides 8 external interrupt lines (IRQ0 - IRQ7) and 8 internal interrupt priority levels (LVL0 - LVL7). External signal can generate interrupt request via the IRQ lines either in "level" mode or "edge" mode. In level mode, the interrupt is generated when the corresponding line is low, in "edge" mode, the interrupt is generated on the falling edge of the signal.

IRQ0 is non-maskable and has the highest priority, followed by LVL0, IRQ1, LVL1, and so on until the lowest priority interrupt level LVL7. To determine which highest priority interrupt to be served, the PowerPC 555 provides a special register, SIVEC, which is also accessible to software interrupt handler. The value stored in SIVEC is a multiple of 4 of the priority of the interrupt.

Exceptions or interrupts are handled partly by assembly and mostly by C code. Assembly code is used for setting/resetting various MPC555 internal registers and saving/restoring CPU states (general purpose and other registers). The C code handles the logic of interrupt handling. When a C function is called within an interrupt handler, a stack is also required.

Sufficiently high update rate of the desired position has to be chosen in order to have a smoothly running motor position control by the PID controllcr. For our application, 1K Hz is adequate. To achieve proper timing, each update has to be triggered by a timer interrupt event. For this purpose, we used the PIT (Periodic Interrupt Timer) facility in the MPC555 chip. When activated, an external interrupt will be generated periodically.

The three motors used in our robot require us to have three PID controllers, each running at an update rate of 1KHz. To achicve this effect, we time sliced the PIT interrupt into a quantum of 333 microseconds. At each PIT interrupt, only one PID is triggered. However, using a very simple round robin triggering mechanism, the three PID controllers seem to run simultaneously by updating all the motor positions once every millisecond.

3.5 Experiments

Test runs have been conducted to determine our robot's capability to walk on various smooth surfaces: plexiglass, painted dry wall, metal surface, and glass, on both single-plane and dual-plane walking. The robot was tested climbing on a vertical surface, walking on a horizontal surface, walking between a horizontal and vertical surfaces (crossing a 90° corner), and walking between specified locations on a vertical surface. During the walking and climbing tests on horizontal and vertical surfaces, respectively, the robot startcd from an initial position and walked a specified number of steps. A sequence of pictures of the robot climbing a vertical surface is shown in Figure 10, Figure 11, and Figure 12. The robot performed these tasks with no difficulties. These were demonstrated at various DARPA PI meetings.

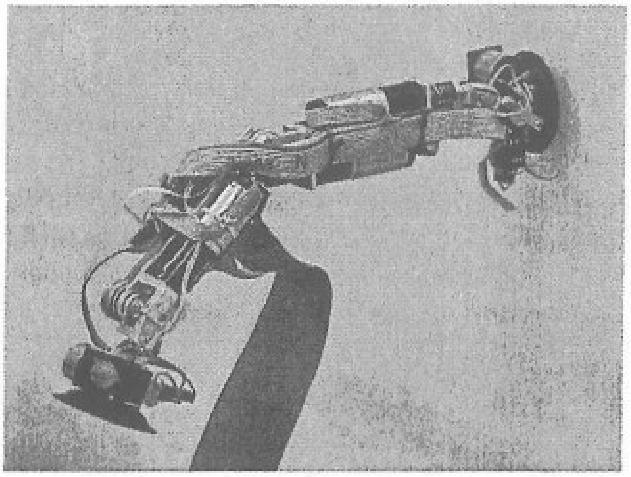


Figure 10. Biped Climbing a Vertical Surface (Start Position and Beginning of Motion)

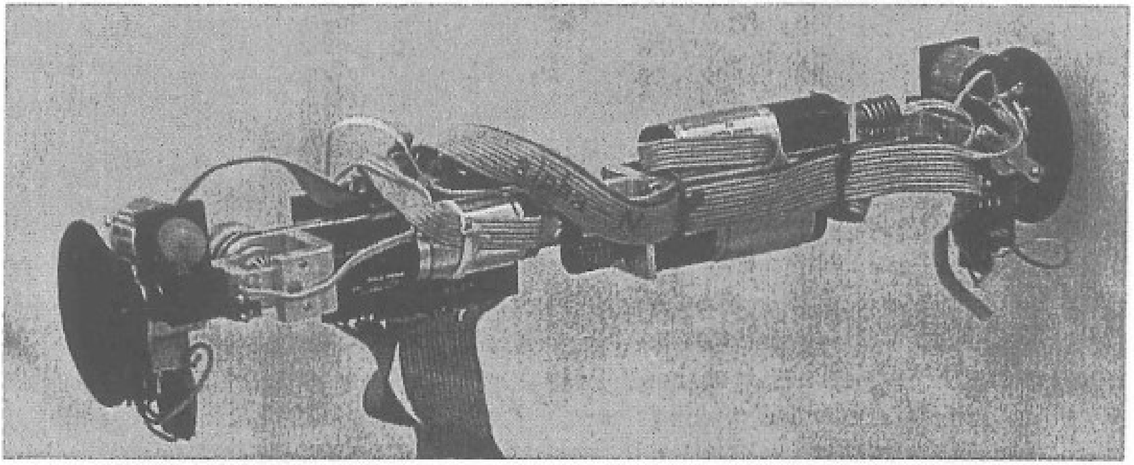


Figure 11. Biped Climbing a Vertical Surface (Halfway Through Step)

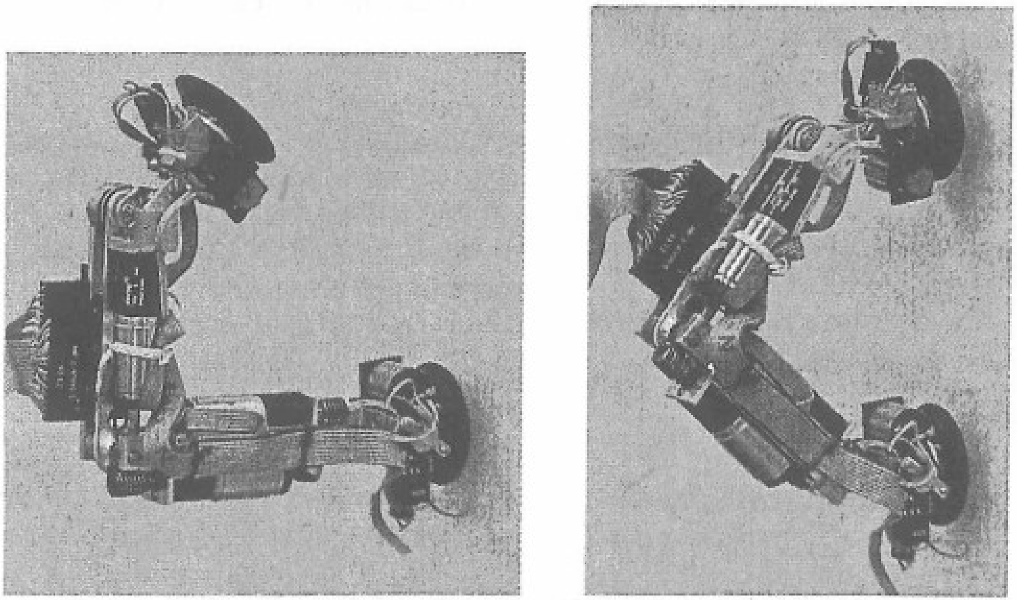


Figure 12. Biped Climbing a Vertical Surface (Near End of Step and Final Position)

3.5.1 Single-Plane Walk

The Flipper has demonstrated its capability to walk on horizontal and vertical surfaces where gravity has different affects on the robot. On horizontal surface, the robot has been tested to walk right side-up as well as upside-down. On walking right side-up, gravity pulls the robot foot towards the surface and the suction rubber did not undergo appreciable deformation. On walking upside-down and on vertical surfaces, gravity pulls the robot away from the surface and the suction rubber undergoes substantial shape deformation.

3.5.2 Dual-Plane Walk

To show its capability to walk across dual planes, we setup two surfaces perpendicular to each other. When the first prototype of the robot did not have touch sensors installed on its feet, the robot had no information of detecting the target surface, and hence the approximate distance to the target surface has to be given in order to solve the inverse kinematics of its joints. After the installation of the touch sensors, the robot obtained this information in real time. By moving the joints in small increments the robot foot can be controlled to touch the target surface and aligned with its normal vector.

The task of crossing from one surface to another consists of several subtasks: approaching the target surface with one suction foot, sensing the surface normal of the target surface, gripping the target surface, and pulling the other foot from the originating surface. When the robot is approaching the target surface, the body of the robot might make a non-perpendicular approach to the surface. In addition, the foot might also make a non-perpendicular angle. These two conditions require a higher-level control algorithm to adjust both the angle of the body (panning) and the foot (tilting) when the robot is in its RFS phase.

Figure 13 (a)-(f) shows a sequence of snapshots of the dual plane walk, where the robot performed the following actions: (a) the robot started in its RFS phase on a vertical wall, (b) using the touch sensors, it aligned its left foot to the target plane (ceiling), (c) suction was established on the left foot, and then released its right foot from the wall, (d) in the LFS phase, the robot continued to walk upside-down (e) suction was established on the right foot (f) it then released the left foot from the surface to continue walking upside-down on the ceiling.

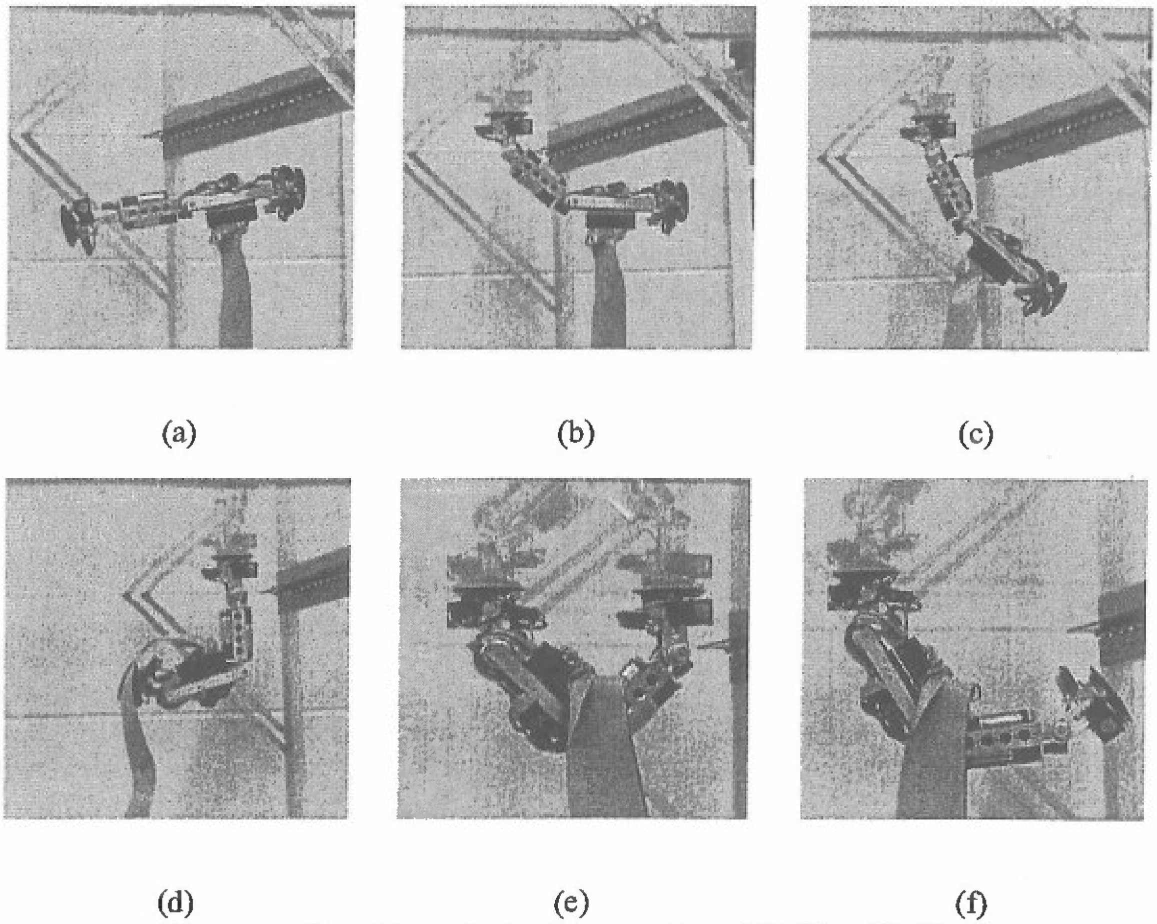


Figure 13. Dual-Plane Walk: Crossing from "Wall" to "Ceiling"

This sequence is considered the most difficult task for the robot due to the following two reasons:

□ Walking across dual planes is more difficult than walking on a single plane
□ The gravitational force exerts the strongest pull when the robot is walking upside-down

4. Miniature Robot With Prismatic Hip (Crawler)

The “Flipper” robot described in the previous section has a biped structure with a revolute hip. Because of its flipping action, it can cover large distances in a relatively short time. However, it cannot travel through confined spaces such as ventilation ducts. This is illustrated in Figure 14. As we can see the flipping motion requires at least an opening of 200mm × 45mm to pass through. For this reason, we have designed a biped miniature robot with prismatic hip joint hereafter called CRAWLER that can remove this restriction. The crawler is designed to pass through a narrow opening of 50mm × 80mm.

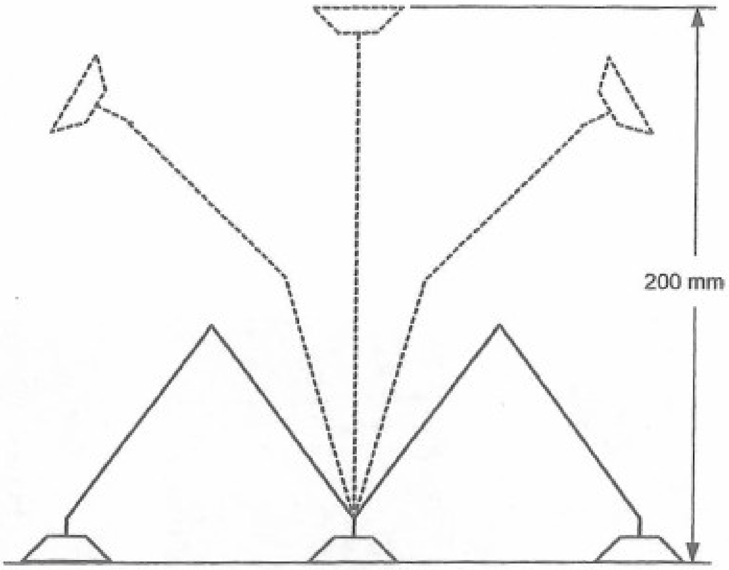


Figure 14. "Flipping" Motion

4.1 Mechanical Structure

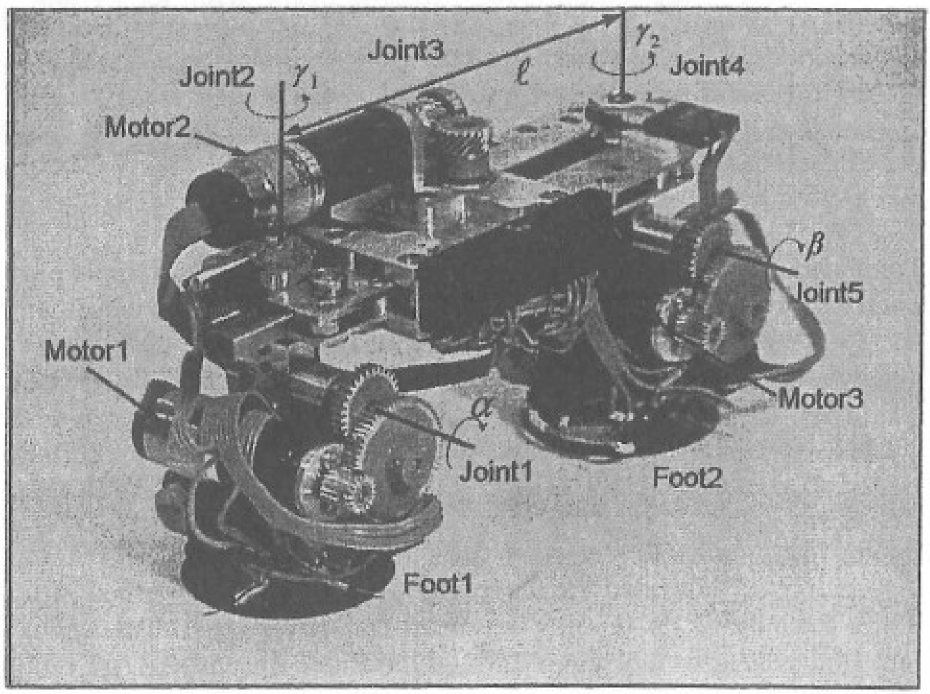


Figure 15. Picture of the Crawler Robot

The crawler robot is shown in Figure 15. The innovative part of the robot design is that we have three coupled joints and hence only three actuators are required for 5 joints. Motor 1 and Motor 3 independently drive joints 1 and 5, respectively; thereby adjusting the tilt angles of the suction foot 1 and foot 2 so that the robot can grip the surface firmly. Motor 2 is responsible for controlling joints 2, 3, and 4. Joint 2 and 4 are revolute joints providing steering capability of the feet relative to the legs. Joint 3 represents the prismatic motion of the legs that allows the robot extending and contracting its legs. The clock-wise (CW) rotation of Motor 2 cause the contraction, i.e., both legs slide into the robot body while the counter-clock-wise (CCW) rotation of the Motor 2 cause the extension, i.e., both legs slide out of the robot body.

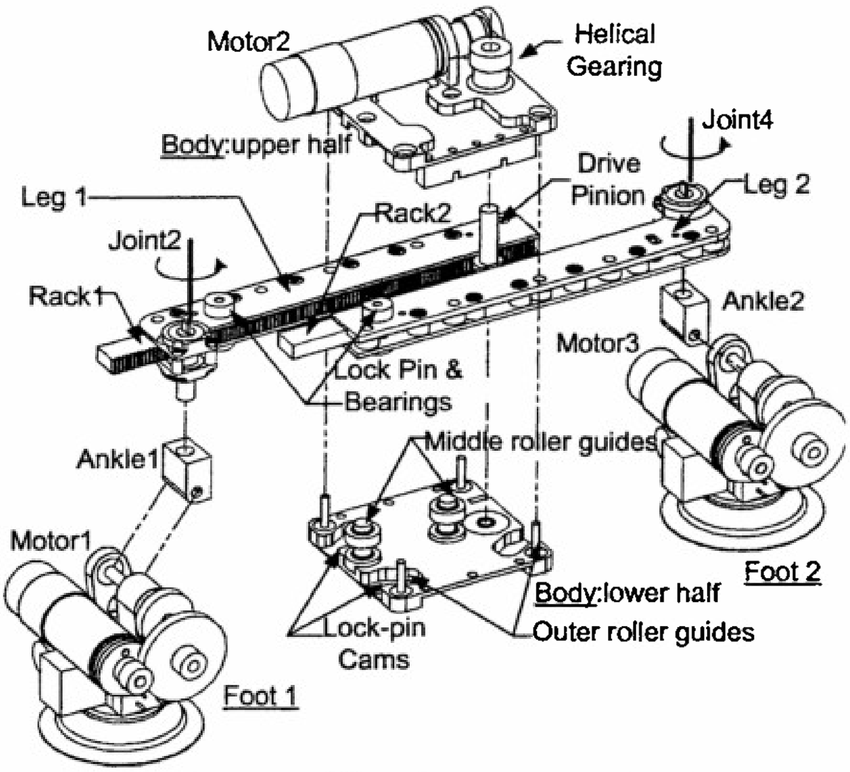


Figure 16. Exploded View of the Crawler Robot

A partially exploded view of the crawler robot is illustrated in Figure 16. The robot consists of identical pairs of legs, racks, ankles and suction feet. Motor 2 drives the helical gear, through the drive pinion, making the two racks sliding in opposite directions. The innovation of the design lies in the structure of the rack/leg pair and the lock-pin cams. Each rack has a notch (see Figure 17). In normal case, the lock pin passes through a slot on the leg and engages the notch by an elastic plate, resulting in the rack/leg pair to slide together. When a rack/leg combination slides in and pulls the lock-pin bearing into the cam slot, the lock-pin cams contained within the upper and lower halves of the body forces the bearing to move along the special curve and push the lock pin outside the notch. This disengaging effect separates the leg/rack pair and prevents the leg from moving but allows the rack to continue sliding. The rack then drives the corresponding robot foot to rotate along joint 2 or joint 4.

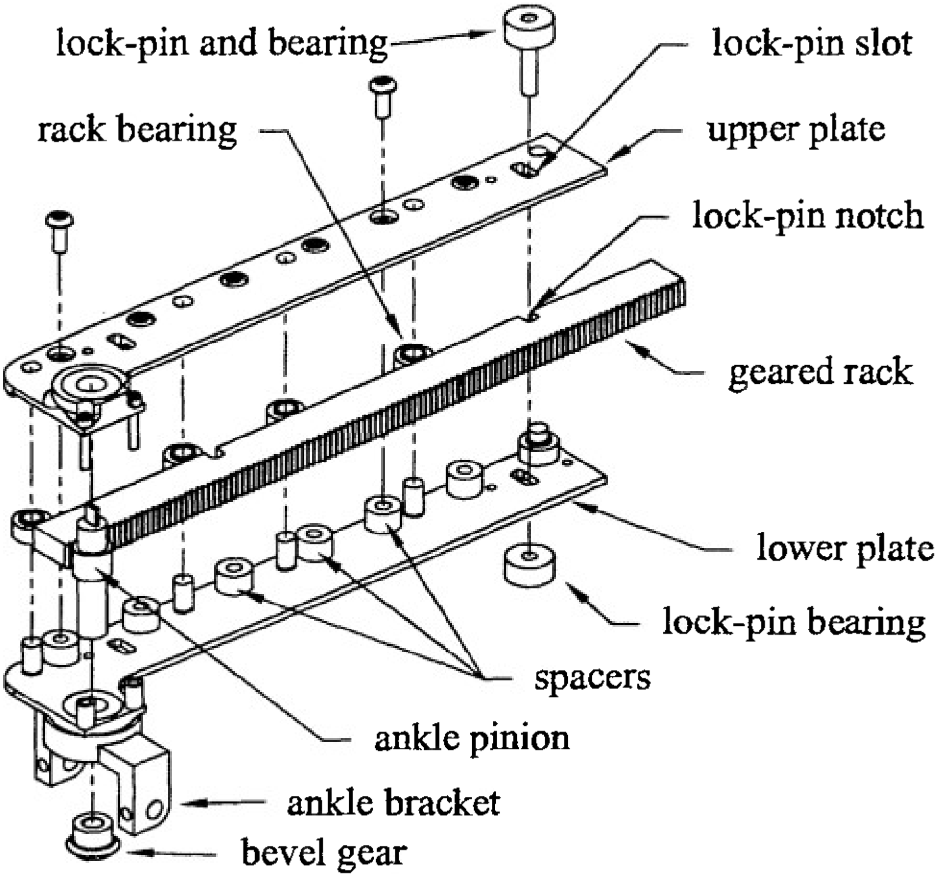


Figure 17. Exploded View of the Crawler Rack/Leg Pair

4.1.1 Locomotion Modes

The crawler microrobot has three motion modes and has the capability to switch between them. Motor 2 drives a set of joints (joint 2, 3, 4) but not all of them simultaneously. In each of the three modes, a particular subset of joints is driven and the remaining joints are locked to prevent rotation. Figure 18 shows the top-down view of the crawler robot and their locomotion modes. Notice that the legs are essentially identical, with the exception of their lock-pin locations. In the case of leg 1, the pin is adjacent to the ankle and enters its cam slot when both legs are contracted. In the case of leg 2, the lock-pin is mounted at the end opposite from the ankle and it enters its cam slot when both legs are extended. The switching between motion modes is achieved by the engaging/disengaging of the lock-pin.

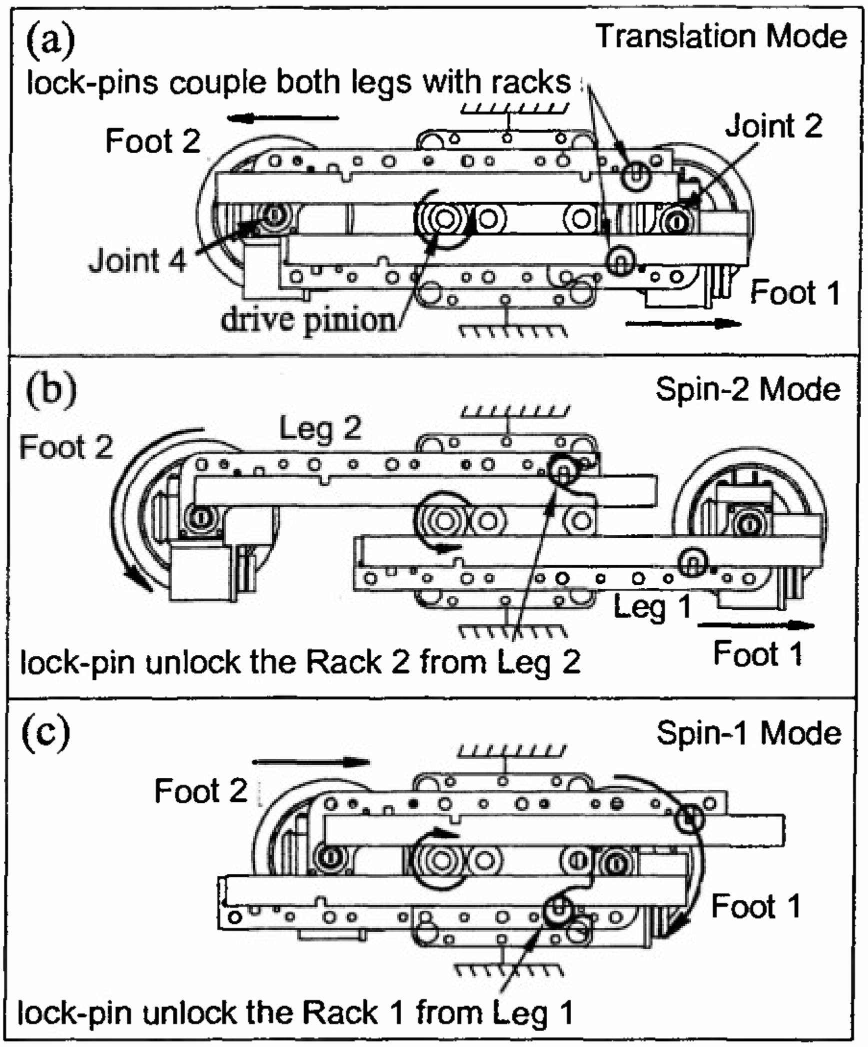


Figure 18. Locomotion Modes of Crawler Robot

Translation Mode: When both the lock-pin bearings are outside the cam, i.e., the legs and their corresponding racks are locked together, joint 2 and 4 are prevented from rotating; and thereby the rotation of the motor 2 causes translation motion of the legs. A counter-clock-wise rotation of the motor 2 causes the legs to extend while clock-wise rotation causes them to contract. If the translation motion continues beyond a certain range, both in extension and contraction, one of the lock-pins will enter its cam slot on the body, causing the mode switch from the translation mode to spin-1 mode or spin-2 mode.

Spin-1 Mode: When lock-pin on leg 1 enters its cam-slot during contraction, it disengages the rack 1 from leg 1 and allows the clock-wise rotation of foot 1 relative to leg 1 about joint 2. Meanwhile, since the lock-pin on leg 2 still couples the leg and rack motion, leg 2 will continue to contract and joint 4 is held fixed.

Spin-2 Mode: If the legs of the robot keep extending in translation mode, the lock-pin bearing on leg 2 will enter its cam slot and unlock the rack 2 from leg 2 causing the counter-clock-wise rotation of foot 2 about joint 4. At mean time, leg1/rack1 pair continues to extend along joint 3 while joint 2 is held fixed.

4.2 Kinematic Model

In robot systems, the control is realized in the joint space, whereas the task level commands are normally expressed in world coordinate space. For the crawler microrobot, the reconnaissance camera is mounted at one of its feet to permit the robot to either look through a glass window or to use the camera like a periscope when the alternative foot supports the robot. The tilt angle of the camera is determined by the position/orientation of the robot free foot related to supporting foot. Thus it is imperative to derive the robot kinematic model which describes the relation between the robot joint variables and the position/orientation of the robot free foot with respect to a fixed reference coordinate frame located at the center of robot supporting foot.

4.2.1 Coordinate Assignment

Because the structure of the crawler robot requires that at least one foot remain in contact with the surface at all times, the setup of the coordinate frames is conducted in the three-dimensional space with respect to right-foot supporting (RFS) phase and left-foot supporting (LFS) phase. In each phase, the robot has two motion modes. One is the translation mode, which means both rack and leg pairs are locked together and thus the middle joint motor 2 drives the two legs sliding in opposite direction. In this mode, only two rotational joints (J1 and J5) and the prismatic joint J3 move. The other is spin mode, which occurs when one of the racks is separated from its leg pair, resulting in the corresponding foot spinning while the other leg/rack pair sliding. In spin mode, two rotational joints (J1, J5), one spin joint (J2 or J4) and the sliding joint (J3) involve in the motion. Since the robot is symmetric, we only analyze the kinematics in LFS phase. The kinematic model is the same for both RFS and LFS phases. The assignment of coordinate frames based on Denavit-Hartenberg (D-H) method [16] for LFS_translation mode is shown in Figure 19.

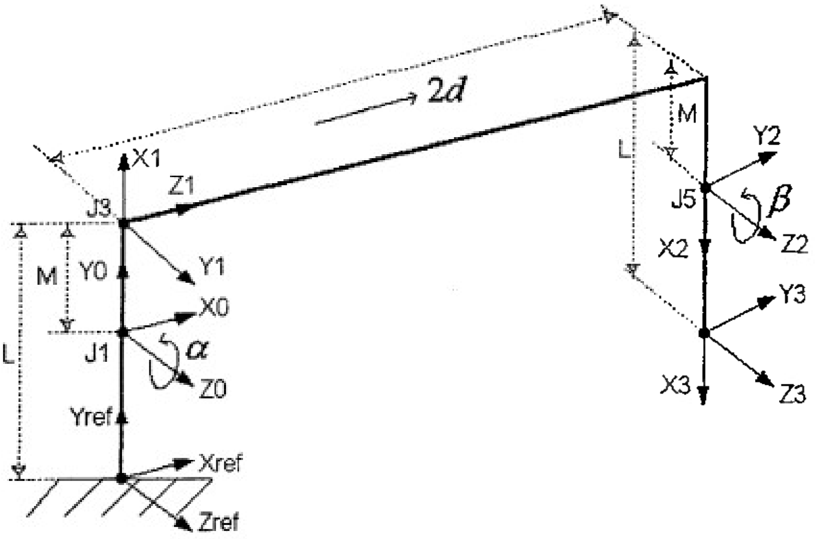


Figure 19. Coordinate Frame: LFS_translation Mode

LFS_translation mode: In LFS_translation mode, the reference frame is attached to the left foot (foot 1) which is fixed on the ground surface. The base coordinate frame is at joint 1 with the Z0 axis aligned with J1 rotation axis. The Z1 and Z2 axes are aligned with the sliding motion axis of J3 and the rotary motion axis of J5 respectively. The right foot (foot2) can move freely with the ``end-effector frame" attached at the center of suction cup. L=68mm is the robot height and M=25mm is the distance between robot ankle to the leg. The prismatic distance between the centers of the robot feet is denoted as *d*.

Table 1-Link coordinate parameters: LFS_translation mode

Motion joint	θ_I	α_i	a_i	d_i
J1 Rotate	α	90°	M	0
J3 Translate	180°	90°	M	*d*
J5 Rotate	β	0	L-M	0

Table 1gives the link coordinate parameters in LFS_translation mode, where α and β and d are joint variables. The four geometric parameters associated with each link in Table 1 are defined as follows:

- θ_i is the joint angle from the X_{i-1} axis to the X_i axis about the Z_{i-1} axis (using the right-hand rule).
- d_i is the distance from the origin of the *(i-1)*th coordinate frame to the intersection of the Z_{i-1} axis with the X_i axis along the Z_{i-1} axis.

- a_i is the offset distance from the intersection of the Z_{i-1} axis with the X_i axis to the origin of the ith frame along the X_i axis (or the shortest distance between the Z_{i-1} and Z_i axes).
- α_i is the offset angle from the Z_{i-1} axis to the Z_i axis about the X_i axis (using the right-hand rule).

LFS_spin mode: The assignment of coordinate frames for LFS_spin mode is shown in Figure 20.

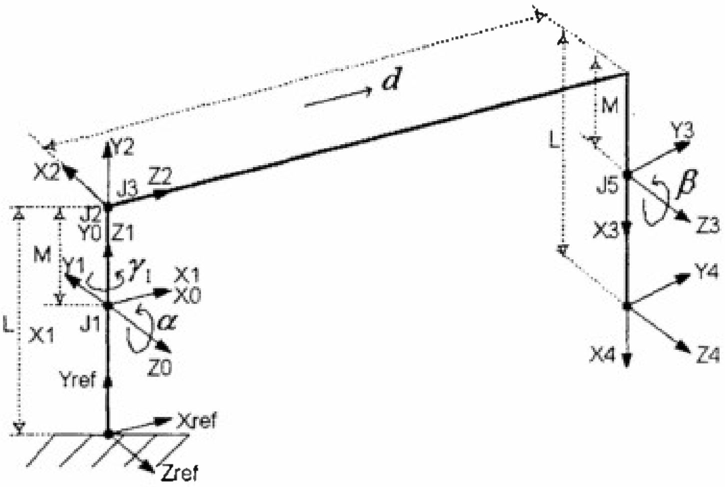


Figure 20. Coordinate Frame: LFS_spin Mode

In LFS_spin mode, four joints involved in the motion. Those are the rotational joint J1 and J5, prismatic slide motion of J3 and spin motion of J2. Note that Z1 is aligned with the spin motion axis of J2. The link coordinate parameters in LFS_spin mode is shown in Table 2, where α, β, γ_1 and d are joint variables.

Table 2-Link coordinate parameters: LFS_spin mode

Motion joint	**θ_i**	**α_i**	**a_i**	**d_i**
J1 Rotate	α	90°	M	0
J2 Spin	γ_1	90°	0	M
J3 Translate	-90°	90°	M	d
J5 Rotate	β	0	L-M	0

In the LFS_spin mode, the slide motion of joint 3 and the spin motion of joint 2 are coupled and are both driven by Motor 2. The prismatic distance leg 2 can move is d, and $d = k\gamma_1$, where k is a constant.

4.2.2 Forward Kinematics

Once the D-H coordinate system has been established for each robot link, a homogeneous transformation matrix ${}^{i-1}A_i$ can easily be developed relating the *i*th coordinate frame to the *(i-1)*th coordinate frame as follows:

$$
{}^{i-1}\mathrm{A}_i = \begin{bmatrix} Cos\theta_i & -Cos\alpha_i Sin\theta_i & Sin\alpha_i Sin\theta_i & a_i Cos\theta_i \\ Sin\theta_i & Cos\alpha_i Cos\theta_i & -Sin\alpha_i Cos\theta_i & a_i Sin\theta_i \\ 0 & Sin\alpha_i & Cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (6)
$$

The homogeneous matrix ${}^{ref}T_i$ which specifies the location of the *i*th coordinate frame with respect to the reference coordinate system is the chain product of successive coordinate transformation matrices of ${}^{i-1}A_i$, and is expressed as

$$
{}^{ref}T_i = {}^{ref}A_0 \, {}^{0}A_1 \, {}^{1}A_2 \ldots {}^{i-1}A_i \qquad (7)
$$

The robot kinematic model is expressed by the 4×4 homogeneous transformation matrix $T={}^{ref}T_n$, *n* is the number of robot moving joints. The *T* matrix has the combined effect of rotation, translation, perspective, and global scaling, and can be expressed as:

$$
\mathrm{T}= \begin{bmatrix} \vec{n} & \vec{s} & \vec{a} & \vec{p} \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} R_{3x3} & P_{3x1} \\ f_{1x3} & 1x1 \end{bmatrix} \qquad (8)
$$

where the upper right submatrix P_{3x1} represents the end-effector position with respect to the reference frame; the upper left submatrix R_{3x3} is the rotation matrix, representing the orientation of the end-effector frame; the lower left submatrix f_{1x3} represents perspective transformation which is useful for computer vision. The vector $\vec{n}$ is the normal vector which is aligned with the direction of X axis of the end-effector frame and is orthogonal to the suction cup surface. The sliding vector $\vec{s}$ is pointing in the direction of the Y axis of the end-effector frame which aligns with the robot slide motion direction. The approach vector $\vec{a}$ is aligned with the direction of Z of the end-effector.

LFS_translation mode: By plugging in the parameter value in Table 1 to Equation (6), the transformation matrices for adjacent coordinate frames in LFS_translation mode can be derived as follows:

$$ {}^{\mathrm{ref}}\mathrm{A}_0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & L-M \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad {}^{0}\mathrm{A}_1 = \begin{bmatrix} Cos\alpha & 0 & Sin\alpha & MCos\alpha \\ Sin\alpha & 0 & -Cos\alpha & MSin\alpha \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (9) $$

$$ {}^{1}\mathrm{A}_2 = \begin{bmatrix} -1 & 0 & 0 & -M \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 2d \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad {}^{2}\mathrm{A}_3 = \begin{bmatrix} Cos\beta & -Sin\beta & 0 & (L-M)Cos\beta \\ Sin\beta & Cos\beta & 0 & (L-M)Sin\beta \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (10) $$

According to Equation (7), the robot kinematics matrix in translation mode is derived as:

$$ T_{translation} = {}^{ref}T_3 = {}^{ref}A_0 \, {}^{0}A_1 \, {}^{1}A_2 \, {}^{2}A_3 = $$

$$ \begin{bmatrix} -Cos(\alpha+\beta) & Sin(\alpha+\beta) & 0 & -(L-M)Cos(\alpha+\beta)+2dSin\alpha \\ -Sin(\alpha+\beta) & -Cos(\alpha+\beta) & 0 & -(L-M)[Sin(\alpha+\beta)-1]-2dCos\alpha \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (11) $$

The rotation submatrix R_{3x3} needs nine elements to completely describe the end-effector orientation of robot. It is necessary to find a convenient expression. A set of Euler angles (φ), (θ), (ϕ), are used to describe the orientation with respect to a fixed reference frame. There are many different types of Euler angle representation [16]. Here we use roll($R_{z,\phi}$), pitch($R_{y,\theta}$), yaw ($R_{x,\varphi}$) to represent the robot orientation. They correspond to the following rotations in sequence:

1. A rotation of φ about the X_{ref} axis ($R_{x,\varphi}$).
2. A rotation of θ about the Y_{ref} axis ($R_{y,\theta}$).
3. A rotation of ϕ about the Z_{ref} axis ($R_{z,\phi}$).

The resultant composite rotation matrix is:

$$ R_{\varphi,\theta.\phi} = R_{z,\phi} \, R_{y,\theta} \, R_{x,\varphi} = $$

$$ \begin{bmatrix} Cos\phi Cos\theta & Cos\phi Sin\theta Sin\varphi - Sin\phi Cos\varphi & Cos\phi Sin\theta Cos\varphi + Sin\phi Sin\varphi \\ Sin\phi Cos\theta & Sin\phi Sin\theta Sin\varphi + Cos\phi Cos\varphi & Sin\phi Sin\theta Cos\varphi - Cos\phi Sin\varphi \\ -Sin\theta & Cos\theta Sin\varphi & Cos\theta Cos\varphi \end{bmatrix} \qquad (12) $$

Comparing this matrix with the rotation submatrix R_{3x3} in Equation (11), we have the following relationship between the joint angle and Euler angle.

$$\begin{cases} \varphi = 0 \\ \theta = 0 \\ \phi = \alpha + \beta - \pi \end{cases}$$

LFS_spin mode: By plugging in the parameter value in Table 2 to Equation (6), the transformation matrices for adjacent coordinate frames in LFS_spin mode can be derived as follows:

$$^{ref}A_0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & L-M \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad {}^0A_1 = \begin{bmatrix} Cos\alpha & 0 & -Sin\alpha & 0 \\ Sin\alpha & 0 & Cos\alpha & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$^1A_2 = \begin{bmatrix} Cos\gamma_1 & 0 & -Sin\gamma_1 & 0 \\ Sin\gamma_1 & 0 & Cos\gamma_1 & 0 \\ 0 & -1 & 0 & M \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad {}^2A_3 = \begin{bmatrix} 0 & 0 & -1 & 0 \\ -1 & 0 & 0 & -M \\ 0 & 1 & 0 & d \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$^3A_4 = \begin{bmatrix} Cos\beta & -Sin\beta & 0 & (L-M)Cos\beta \\ Sin\beta & Cos\beta & 0 & (L-M)Sin\beta \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (13)$$

According to Equation (7), the robot kinematics matrix in LFS_spin mode is derived as:

$$T_{spin} = {}^{ref}T_4 = {}^{ref}A_0\,{}^0A_1\,{}^1A_2\,{}^2A_3\,{}^3A_4 =$$

$$\begin{bmatrix} Sin\alpha Cos\beta + Cos\alpha Sin\beta Sin\gamma_1 & -Sin\alpha Sin\beta + Cos\alpha Cos\beta Sin\gamma_1 & -Cos\alpha Cos\gamma_1 & p_x \\ -Cos\alpha Cos\beta + Sin\alpha Sin\beta Sin\gamma_1 & Cos\alpha Sin\beta + Sin\alpha Cos\beta Sin\gamma_1 & -Sin\alpha Cos\gamma_1 & p_y \\ Sin\beta Cos\gamma_1 & Cos\beta Cos\gamma_1 & Sin\gamma_1 & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} (14)$$

where

$p_x =$ (L-M)($Sin\alpha\ Cos\beta + Cos\alpha\ Sin\beta\ Sin\gamma_1$) + $d\ Cos\alpha\ Sin\gamma_1$

$p_y =$ (L-M)(1- $Cos\alpha\ Cos\beta + Sin\alpha\ Sin\beta\ Sin\gamma_1$) + $d\ Sin\alpha\ Sin\gamma_1$

$p_z =$ (L-M) $Sin\beta\ Cos\gamma_1 + d\ Cos\ \gamma_1$

$d = k\gamma_1$

Thus far, the robot transformation matrix has been deduced for both motion modes. The solution of forward kinematic problem is obtained by evaluating each element in T matrix for a given set of joint variables.

4.2.3 Inverse Kinematics

In this section, the inverse kinematics are derived. In the motion planning and control of the robot, the orientation vectors play a critical role. In order to make the suction foot grip the surface firmly, we must ensure that the normal vector $\vec{n}$ be perpendicular to the contact surfacc. By specifying the position vector $\vec{p}$ and the foot normal vector $\vec{n}$, the relation between the suction foot and the contact surface is determined. Therefore, it will be convenient in the implementation if the solution of inverse kinematics only contains the two vectors $\vec{n}$ and $\vec{p}$.

In order to evaluate θ for $-\pi \le \theta \le \pi$, an arc tangent function, $a\tan 2(y,x)$, which returns $\tan^{-1}(y/x)$ adjusted to the proper quadrant is defined and will be used in the solution of inverse kinematics.

$$\theta = \text{atan2(y, x)} = \begin{cases} 0° \le \theta \le 90° & for \quad +x \quad and \quad +y \\ 90° \le \theta \le 180° & for \quad -x \quad and \quad +y \\ -180° \le \theta \le -90° & for \quad -x \quad and \quad -y \\ -90° \le \theta \le 0° & for \quad +x \quad and \quad -y \end{cases} \tag{15}$$

Inverse kinematics analysis is considered for both LFS_translation and LFS_spin modes.

LFS_translation mode: The transformation matrix is rewritten as in the following.

$$\text{T} = \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$

$$\begin{bmatrix} -Cos(\alpha+\beta) & Sin(\alpha+\beta) & 0 & -(L-M)Cos(\alpha+\beta)+2dSin\alpha \\ -Sin(\alpha+\beta) & -Cos(\alpha+\beta) & 0 & -(L-M)[Sin(\alpha+\beta)-1]-2dCos\alpha \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{16}$$

By comparing both sides of this Equation, we have:

$$\frac{n_y}{n_x} = \frac{Sin(\alpha+\beta)}{Cos(\alpha+\beta)} \Rightarrow \alpha+\beta = \arctan(n_y, n_x) \tag{17}$$

$$\begin{cases} 2dSin\alpha = p_x - (L-M)n_x \\ 2dCos\alpha = -p_y + (L-M)(n_y+1) \end{cases} \Rightarrow \begin{cases} \tan\alpha = \dfrac{p_x - (L-M)n_x}{-p_y + (L-M)(n_y+1)} \\ d = \dfrac{p_x - (L-M)n_x}{2Sin\alpha} \end{cases} \qquad (18)$$

From Equation (17) and Equation (18), the joint variables are solved using position vector $\vec{p}$ and the foot normal vector $\vec{n}$:

$$\begin{cases} \alpha = a\tan 2[p_x - (L-M)n_x, -p_y + (L-M)(n_y+1) \\ \beta = a\tan 2[n_y, n_x] - \alpha \\ d = \dfrac{p_x - (L-M)n_x}{2Sin\alpha} \end{cases} \qquad (19)$$

LFS_spin mode: By analyzing the Equation (14), we have:

$$\begin{cases} p_x = (L-M)n_x + dCos\alpha Sin\gamma_1 \\ p_y = (L-M)(1+n_y) + dSin\alpha Sin\gamma_1 \end{cases} \Rightarrow \tan\alpha = \frac{p_y - (L-M)(n_y+1)}{p_x - (L-M)n_x} \qquad (20)$$

$$\begin{cases} p_x = (L-M)n_x + dCos\alpha Sin\gamma_1 \\ p_z = (L-M)n_z + dCos\gamma_1 \end{cases} \Rightarrow \tan\gamma_1 = \frac{p_x - (L-M)n_x}{(p_z - (L-M)n_x)Cos\alpha} \qquad (21)$$

$$\begin{cases} n_z = Sin\beta Cos\gamma_1 \\ s_z = Cos\beta Cos\gamma_1 \end{cases} \Rightarrow \tan\beta = \frac{n_z}{s_z} \qquad (22)$$

From Equations (20), (21) and (22), the joint variables are solved as follows:

$$\begin{cases} \alpha = a\tan 2[p_y - (L-M)(n_y+1), p_x - (L-M)n_x \\ \gamma_1 = a\tan 2[p_x - (L-M)n_x, (p_z - (L-M)n_z)Cos\alpha \\ d = k\gamma_1 \\ \beta = a\tan 2(n_z, s_z) \end{cases} \qquad (23)$$

So far, the robot inverse kinematics has been derived for both motion modes. The solution of the inverse kinematics will be used in the implementation of robot motion control.

4.3 Controller Hardware Design

As a self-contained embedded system, the crawler robot needs to carry its own power source, sensors, control system, and associated hardware. Thus minimization of weight and power consumption is critical to prolonged operation. The TMS320LF2407 digital signal processor (DSP) from Texas Instruments (TI) Inc. is an ideal candidate for an embedded controller because of its high-speed performance, its support for multi-motor control, and its low power consumption. This section describes the control system design and implementation based on a TI DSP.

4.3.1 Actuators and Sensors

To minimize weight and power consumption, the crawler robot is designed with limited number of actuators and sensors. The actuators include three DC servo motors with encoder feedback, two suction pump motors, and two micro valves. The primary sensor components include two pressure sensors and six touch sensors located at the suction feet. In order to distinguish between different motion modes, two contact switches are installed on robot legs to determine whether the leg and rack are locked. For tele-operation, the robot can also be controlled with a remote control unit shown in Figure 9. All of the signals from those components and sensors need to be processed and integrated into the on-board control system. The crawler robot can carry additional sensors, such as wireless cameras, sonar sensors, inclinometers, microphones and digital infrared sensors depending on the applications. Table 3 shows the major components of the robot and their functions.

Table 3-Actuators and sensors used in Crawler micro-robot

Actuators and sensors	Function
DC Servo Motor 1	Drives joint 1; adjust tilt angle of foot 1
DC Servo Motor 2	Drives joints 2, 3, 4; but not all of them simultaneously
DC Servo Motor 3	Drives Joint 5; adjust tilt angle of foot 2
Encoder 1, 2, 3	Provide joint position information
Suction pump 1, 2	Create suction force, support foot on surfaces
Micro-valve 1, 2	Release the suction foot
Pressure sensor 1, 2	Decide whether the suction foot is firmly attached to a surface
Touch sensors	Attached to the suction cup in different radial directions; facilitate the adjustment of the suction foot orientation.
Contact switch1,2	Determine if the leg/rack pair is engaged; distinguish between different motion modes,
Transmitter/Receiver	Wireless communication between the robot and its 16-key remote control unit
Wireless camera	Get images and transmit them to a host computer

4.3.2 Control System Structure

The control system structure of the robot is shown in Figure 21. The physical actuators and sensors are represented in the right block. Other blocks represent the on-board software modules including command interpreter, task level scheduler, trajectory planner, joint level controller and motion planner. The user commands, such as "move forward" "make left turn" are transmitted from the remote control unit held by a human operator and decoded by the on-board command interpreter. The generated task level commands are then fed into the task level scheduler. The task level scheduler uses a finite state machine to keep track of robot motion status and decompose the command into several motion steps. The trajectory planner solves the inverse kinematics model and interpolates the path to generate a set of desired joint angles. The digital motor controller then drives each joint to the desired set points so that the foot is placed to the desired location.

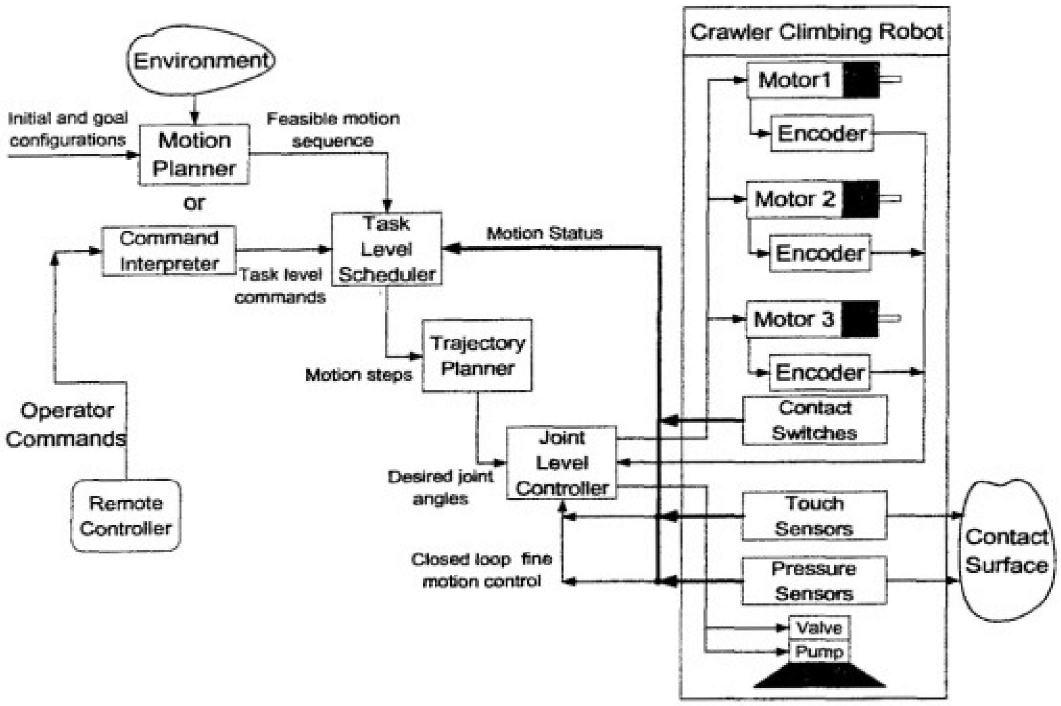


Figure 21. Control System Block Diagram of the Crawler Robot

Placing the foot on a surface consists of gross and fine motion control. Gross motion control is based on the desired solution of the inverse kinematics and it brings the free foot to the neighborhood of the desired position and orientation. However, the results of the actual foot placement may not be sufficiently accurate due to the existence of uncertainties and disturbances caused by backlash, gear friction, sensor error, and varying gravitational effects. Thus, after the gross motion control, the joint level controller conducts fine motion control. Fine motion control utilizes the feedback signal from touch sensors and pressure sensors to adjust the robot foot to ensure that it grips the contact surface reliably. Adjustment of the robot foot is a closed loop control process and only requires varying the tilt angle in small increments to align the foot with the contact surface.

When the robot has motion planning ability, the motion planner generates a feasible motion sequence and transmits it to the task level scheduler. After the motion sequence has been executed, the robot is able to travel from its initial configuration to its goal configuration, whilc avoiding the obstacles in the environment.

4.3.3 TI LF2407 DSP Overview

The TMS320LF2407 device [17] is a new member of the C2000 family of TI DSP controllers. It is targeted to meet thc needs of control system applications. By integrating the high performance of a DSP core and the on-chip peripherals into a single-chip solution, the LF2407 device is a low-cost alternative to traditional microcontroller units (MCUs) and expensive multi-chip designs. This chip provides all the resources needed to build a self-contained embedded control system. The functional block diagram of the LF2407 DSP controller is shown in Figure 22.

Thc LF2407 DSP controller is based on the 16-bit, fixed-point, low-power C2xx DSP core. The parallel architecture of the central processing unit (CPU) enables the DSP to perform in high speed at 30 million instructions per second (MIPS). Thus it allows the DSP to often process algorithms in real time rather than approximate results with look-up tables.

By integrating memory and peripherals onto a single chip, the C2000 family DSP devices reduce system cost and save circuit board space. LF2407 DSP provides 32K words on-chip flash memory, which offcrs a re-programmable solution useful for the initial prototyping of applications. The 2.5K words on-chip random access memory (RAM) includes 544 words of Dual-Access RAM (DARAM) and 2K words of Single-Access RAM (SARAM), which are configurable as data or program memory. With the help of an external memory interface, the LF2407 chip can offer up to 64K program memory and 64K data memory space.

The LF2407 DSP has two event manager modules (EVA and EVB) that have been optimized for digital motor control. Capabilitics of each modulc include two 16-bit general-purpose timers, three capture units, and eight 16-bit pulse-width modulation (PWM) channels. Two of the capture units have built-in quadrature encoder pulse (QEP) circuits, which are used to obtain the motor encoder readings easily.

The high performance analog-to-digital converter (ADC) has 10-bits resolution, a minimum conversion time of 500 ns, and up to 16 channels of analog input. The auto-sequencing capability of the ADC allows a maximum of 16 conversions to take place in a single conversion session without any CPU overhead.

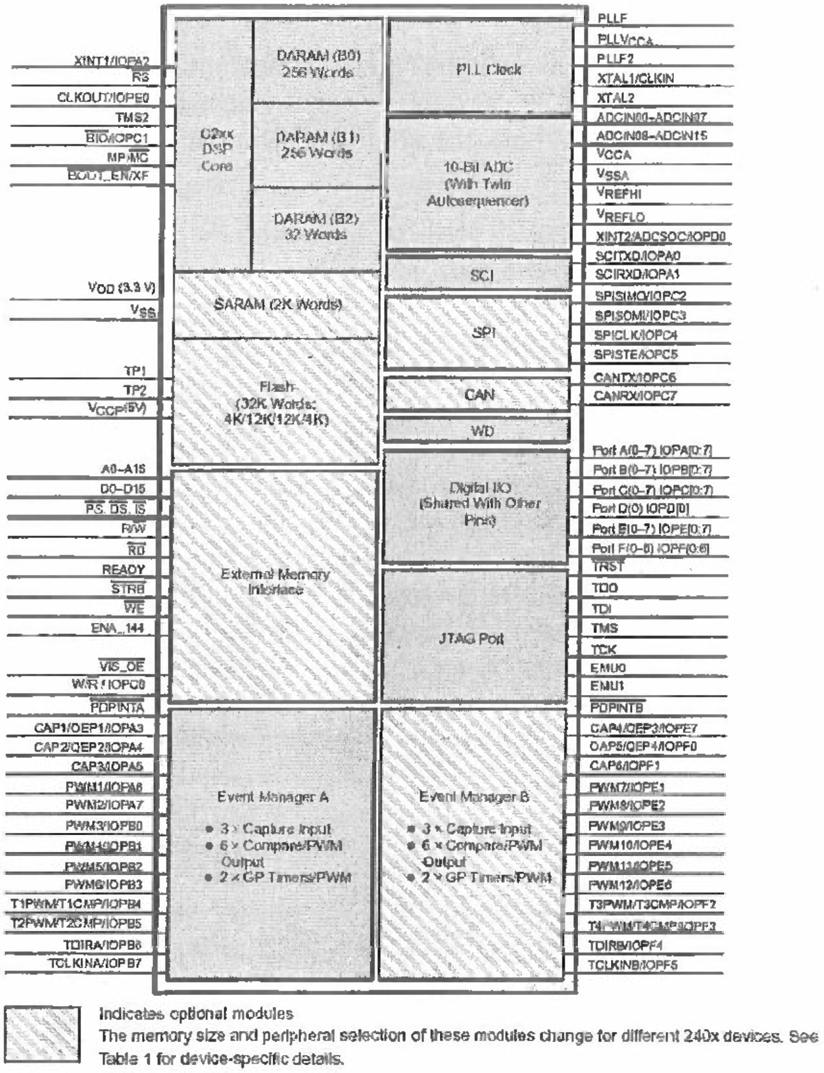


Figure 22. Functional Block Diagram of the LF2407 DSP Controller

A serial communications interface (SCI) is integrated on the device to provide asynchronous communication capability. For systems requiring additional communication interfaces, the LF2407 offers a synchronous serial peripheral interface (SPI) and a controller area network (CAN) communications module. To maximize device flexibility, functional pins are configurable as general purpose inputs/outputs (GPIO). Thus, up to 40 multiplexed digital I/O pins are available. JTAG module provides non-intrusive real-time debugging capability which is helpful to reduce the system development time.

The LF2407 DSP is based on low-power 3.3V CMOS technology and it integrates many power management features. It has three power down modes and the ability to power-down each peripheral module independently. These features make the LF2407 DSP a desirable device to be used in an embedded control system with power constraints.

4.3.4 DSP Implementation of the Controller

Due to the limitations on size, wcight and power consumption, we must efficiently utilize the resource of DSP chip and use as fewer components as possible when designing the control system. Figure 23 illustrates the controller block diagram based on a TI LF2407 DSP chip.

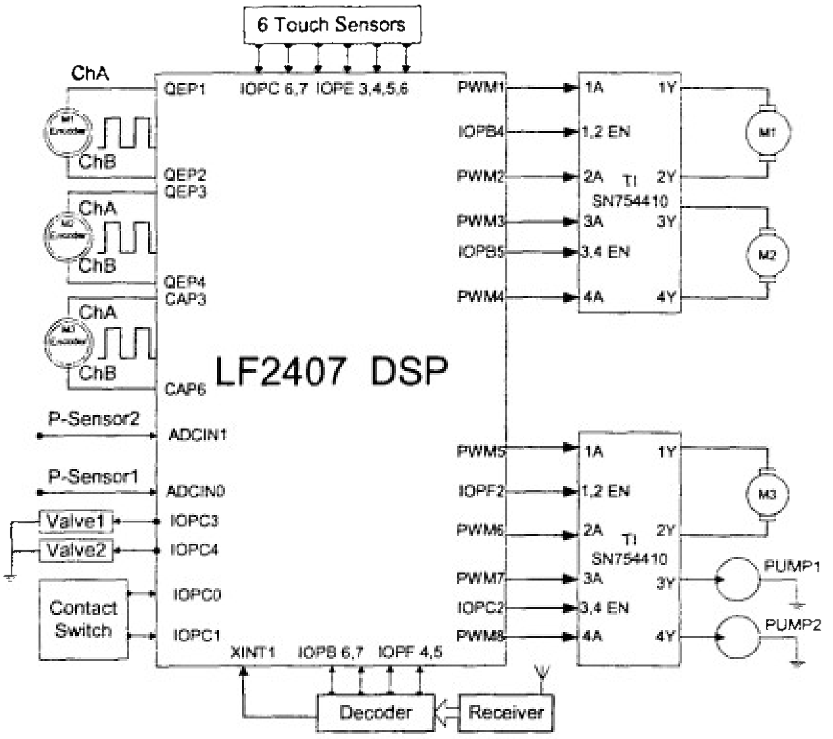


Figure 23. Block Diagram of the DSP-based Controller

Two quadruple half-H driver SN754410 from TI are used to drive the three servomotors and the two pump motors. The servomotors are driven by the PWM outputs PWM1, 2, PWM3, 4, and PWM5, 6 of EVA via the H-bridge driver. Timer 1 is used as the time base to generate the PWM signal with a frequency of 20KHz. Timer3 is used to generate the servo-control sampling rate of 1KHz. During each servo sampling period (1ms), the compare registers, CMPR1, CMPR2, and CMPR3 are updated for Motor1, Motor2 and Motor3, respectively according to the calculated PID control value. LF2407 DSP has two built-in quadrature encoder pulsc (QEP) circuits. The encoder readings of servo Motor1 and Motor2 are easily obtained using the QEP1/QEP2 and QEP3/QEP4 of the Event Manager with Timer2 and Timer4 as the time base, respectively. However, for Motor3, we need to employ alternative method to get the encoder reading. The encoder channel A of servo Motor3 is connected to CAP3 pin and the capture unit 3 is enabled to detect the rising edge of the encoder pulse. Channel B is connected to the CAP6/IOPF1 pin and this pin is configured as shared pin function of digital input IOPF1. Whenever the capture interrupt is triggered in CAP3, the interrupt service routine detects the digital level of thc encoder channel B to determine the rotation direction and adjusts the encoder pulse count value. Apart from serving as the time base for PWM waveform, Timer1 is also used as the time base for capture 3 operation. These two functions do not affect each other. A software solution for getting the encoder reading of Motor 3 using the capture unit is described in section 4.4.

For the two pump motors, we only need a binary switch. So we configure PWM7/IOPE1 and PWM8/IOPE2 as digital output pins. With the H-bridge driver, we can simply turn on the pump by setting the output as high. The micro valves are also controlled by the digital outputs from DSP via two transistors as the drivers.

The touch sensors consist of an outer tube and an inner super-elastic wire isolated by a silicon tube. When the inner wire touches the outer tube the switch closed. In such a way, the touch sensors provide a set of digital inputs to DSP to facilitate the robot in adjusting the suction foot orientation. The contact switches produce high-level or low-level digital signals that are used to determine the motion mode switching and are connected directly to the I/O pins of the DSP. The analog inputs from the pressure sensors are converted by the ADC to determine whether the suction foot is securely attached to a flat surface. A receiver module and a decoder chip are used for remote control operation. The impulse signal from the decoder triggers the external interrupt pin of the DSP. The interrupt service routine processes the four digital I/O outputs from the decoder and translates the remote control signal into proper commands.

By efficiently utilizing the resources of the DSP chip, the component count of the embedded controller is minimized and the control system becomes self-contained.

4.4 Controller Software Development

4.4.1 Software Modules

Software modules have been developed in C and TI DSP assembly languages to control the climbing robots. They work well for both Flipper and Crawler robots. The main software modules running on the embedded controller are:

1. Task level scheduler
2. Trajectory planner
3. Joint level PID servo control
4. Remote command interpreter
5. Communication module

Task level scheduler detects the motion status and decomposes the task level commands into several motion steps. A finite state machine is developed to keep track of the robot motion status, such as the switching between different motion modes, the changing of standing foot, and the states of the suction pumps.

Trajectory planner activates different kinematic models according to different motion modes and solvcs the invcrse kinematics to plan the joint level trajectory. The interpolation is conducted to generate desired joint angles.

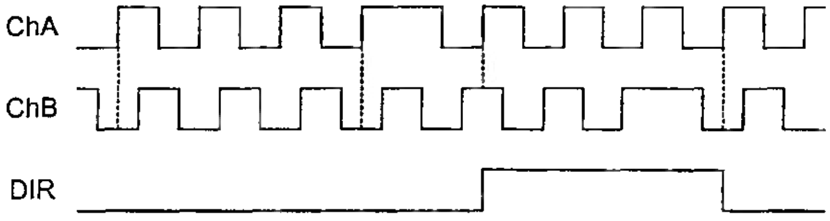
PID servo control is implemented for each servo motor to drive the motor to the desired angle.

Command interpreter is used to decode the commands sent by a human operator through a remote transmitter. The receiver chip on the embedded controller board will trigger an external interrupt whenever a button in thc remote controller panel is pushed. The interrupt service routine processes four digital I/O outputs from the receiver and translates the remote control signal into proper commands.

Communication module handles the RS-232 serial communication between embedded controller board and a host computer. A command interface is implemented to facilitate the testing of the robot control systcm.

4.4.2 Encoder Reading

Accurate digital motor control is the basic requirement for the robot to accomplish certain tasks. To achieve this, the servo controller requires position information from the encoder as feedback. As mentioned before, it is easy to obtain encoder reading for motor 1 and motor 2 using the two build-in QEP circuit of DSP. However, for motor 3, software solution is adopted to get the encoder reading because adding additional circuit is not desirable for micro-robot limited in size and weight.



Figurc 24. Encoder Pulse and Motor Drive Directions

The analysis of the magnetic encoder waveform of the servomotor shown in Figure 24 indicates that:

1. The phase shift of the encoder pulse sequences between two channels (A and B) is 90 degrees.
2. The logic levels of channel B corresponding to the rising edges of channel A alternate when the motor changes direction.

These properties are utilized to obtain the encoder reading. The encoder channel A of servo Motor3 is connected to CAP3 pin and the capture unit 3 is enabled to detect the rising edge of the encoder pulse. Channel B is connected to input digital I/O pin IOPF1. Whenever a rising edge is detected, the capture interrupt is triggered. In the interrupt service routine, the logical level of channel B is detected through IOPF1. When channel B is logic low, it means that the motor rotates in positive direction, then the encoder pulse count variable *CAP3_cnt* adds one. When channel B is logic high, it means the motor has changed the rotation direction, then *CAP3_cnt* minus one. In such a way, *CAP3_cnt* updates its value by +1 or -1 according to the rotation direction every time the encoder pulse triggers. It is clear that the value of *CAP3_cnt* indicates the motor angle expressed in encoder counts. *CAP3_cnt* is set as a global variable and its value can be accessed to accomplish feedback control.

4.4.3 Overflow/underflow Adjustment

Since the register of LF2407 DSP is 16 bits, it is not enough to hold the motor position value in the operation range. Special care must be taken to deal with overflow/underflow of encoder reading. When the difference between the encoder reading of two consecutive samples is unreasonably large, the overflow/underflow is detected. The *wrap_correct* variable is updated to handle the overflow/underflow. The actual motor position is calculated by add the *wrap_correct* value and the encoder count value *encoder_cnt*. The sample code is shown as follows:

```
void Servo_PID(channel)
{long velocity[3],encoder_cnt[3],position[3],wrap_correct[3];
...
/*    Overflow/underflow adjustment */
velocity[channel]=-encoder_cnt(channel); /* Encoder of last sample */
encoder_cnt[channel]=GetEncoder(channel);/* Encoder of current sample
*/
velocity[channel]+=encoder_cnt[channel];/* Calculate velocity */
if (labs(velocity[channel])>ox7FFF) {     /* If overflow/underflow*/
      if (velocity[channel]&0x8000 {            /* If negative */
            velocity[channel]-=0x10000;
            wrap_correct[channel]-=0x10000;}
      else {      velocity[channel]+=0x10000;   /* If positive */
            wrap_correct[channel]+=0x10000}
...
}
position[channel]=encoder_cnt[channel]+wrap_correct[channel];
}
```

4.4.4 PID Controller

PID control is the fundamental module for the robot control system. A specific interrupt service routine (ISR) processes the PID control and motion profile calculation. Timer 3 is selected to generate servo control sampling period as 1 ms. During each sampling period, 1SR read motor position, calculate new trajectory point according to the desired motion profile, update the position error and calculate P1D controllcr output and then set PWM duty cycle. Those tasks arc interleaved for each of the active motors, i.e. the motors can be controlled simultaneously as long as they are needed and sct active.

For smooth motion control, a motion profile is necessary to control the motor acceleration and deceleration. When a motor is commanded to rotate a certain angle, the desired position for each servo step must be calculated by the motion profile routine. Without the motion profile, motion will be abrupt, causing excessive wear on the mechanical components and degrading the performance of the control algorithm. For our application, a linear piecewise trapezoidal/triangular velocity is implemented. For detailed description of software modules, the readers are referred to [18].

4.5 Motion Planning

Due to the under-actuated structure and the different motion modes of the robot, the motion planning becomcs very important for robot to accomplish a certain task. The key point in the motion planning is to determine which motion modes should be applied to and combined together to accomplish the task. In order to distinguish between different motion modes, we implement a contact switch on each leg to determine whether the leg and rack are locked together. A **finite state machine** is employed to keep track of the robot motion states so that different kinematic model equations are used to schedule the robot gait.

Table 4-Finite state machine fields

ENDR	ENDT	PUMP1	PUMP2	SW2	SW1

PUMPi=1, suction pump of foot i is on, foot i is the standing foot;
PUMPi=0, suction pump of foot i is off, foot i is the free foot.

SWi=1, Contact switch i is on,
SWi=0, Contact switch i is off.
When SW1=SW2=0, the robot is in translation motion mode.
When SW1=1, SW2=0, the robot is in spin-1 mode.
When SW1=0, SW2=1, the robot is in spin-2 mode.

ENDR=1, the leg reach the end of a spin motion mode,
ENDR=0, the leg is in the starting point of the spin motion mode.

ENDT=1, the leg reach the end of a translation motion mode,
ENDT=0, the leg is in the starting point of the translation motion mode.

For example, in order to accomplish "move forward" task, the robot will essentially operate in RFS_translation and LFS_translation motion modes. The task scheduler generates a sequence of the joint motions to accomplish one step of crawling forward. Assuming foot 1 grips the ground initially, the robot will first lift foot 2 off the surface through articulation of its ankle joint 1. The robot will then extend along joint 3 and then articulate joint 1 to bring foot 2 in contact with the surface. At this time, the pump of foot 2 will be turned on and the fine adjustment of the tilt angle of foot 2 will be performed according to the pressure sensor and touch sensor information to ensure foot 2 grips the ground firmly. After that, the foot 1 suction will be released and the robot be supported by foot 2, the same procedure will be repeated under RFS_slide mode to walk along a straight line with the gait resembling the motion of an inchworm. Since the robot operates only in translation mode, the kinematic Equations (11) and (19) are used to generate joint level trajectory.

If the robot needs to steer its direction, both translation and spin motion modes will be employed. Assuming the robot is initially supported by foot 1 and in LFS_translation mode, the robot will lift up its body by articulating joint 1 and then contract its body along joint 3. It will continue to contract till the lock-pin on leg 1 enters its cam slot and switches the motion mode to LFS_spin. The robot will now begin to swing on foot 1 in the clock-wise direction. After reaching the desired turning angle, the robot will lower its body and anchor foot 2 on the surface. The kinematic Equations (11), (19), and Equations (14), (23) are used to generate joint level trajectory for translation mode and spin mode, respectively.

4.6 Experiments

The on-board embedded controller based on TMS320LF2407 DSP chip has been designed and implemented. The experiments were conducted to evaluate the performance.

Figure 25 shows a sequence of snapshots of the Crawler walking around a corner of a maze. Figure 25(a-d) shows that the Crawler makes a 45° left turn. Then it moves forward by two steps (e-h) and makes second 45° left turn (i,j) and then continues moving forward (k,l).

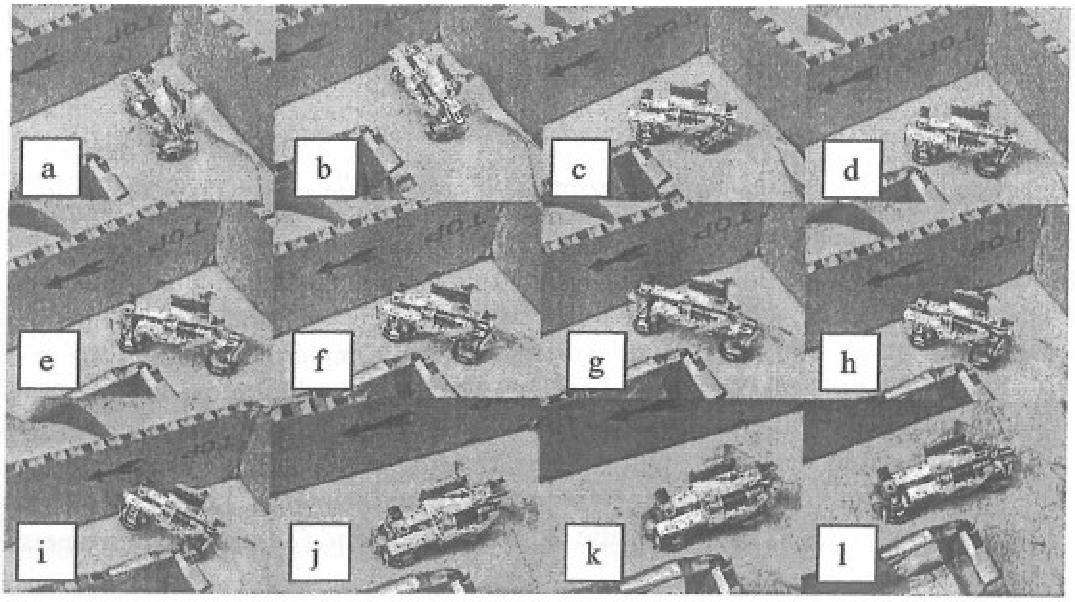


Figure 25. Crawler Walking Around a Corner

Figure 26 shows a sequence of snapshots of the Crawler climbing a vertical wall.

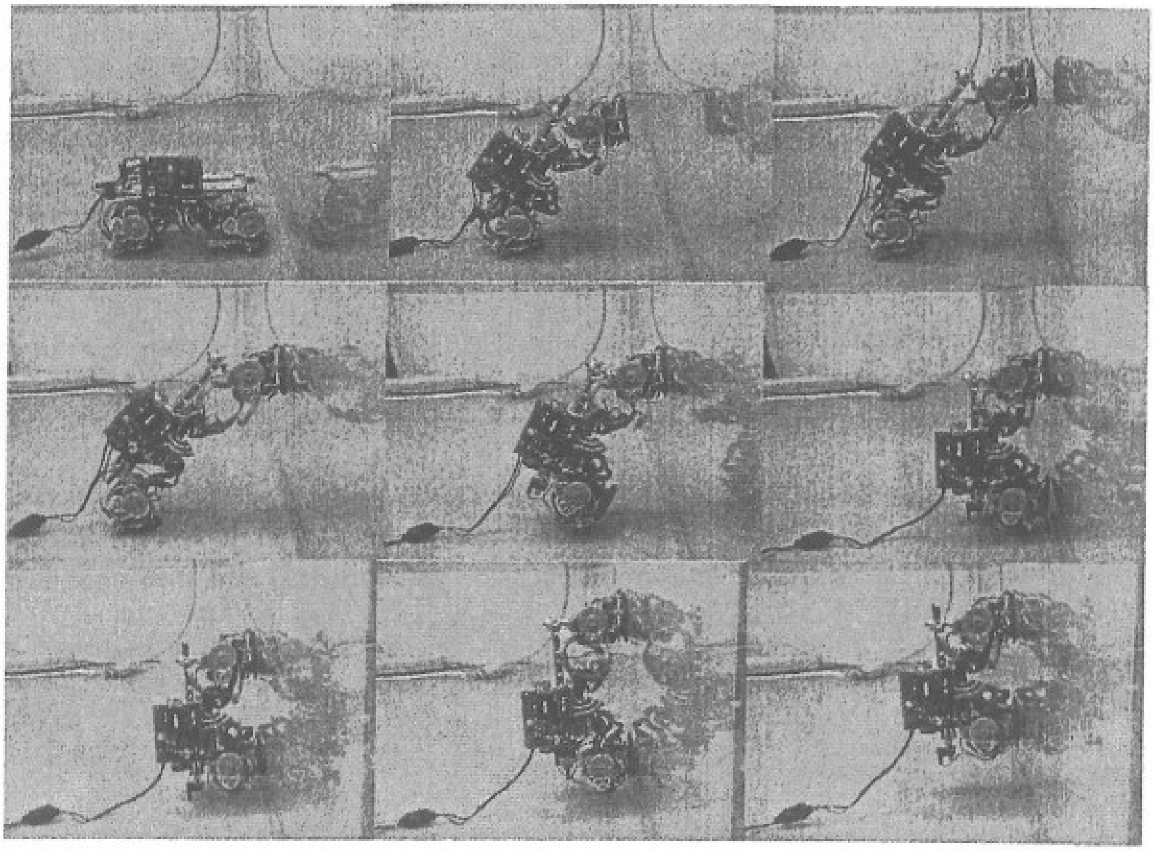


Figure 26. Crawler Climbing from a Floor to a Wall

5. Conclusions and Recommendations

In this report, the mechanical structure, the kinematics model, and the control implementation of two wall-climbing micro-robots are discussed. These robots have the capability to walk on both horizontal and vertical surfaces, and transit between some inclined surfaces. We chose the under-actuated mechanical structure to reduce the weight, power consumption; however the structure increases the control complexity. Micro-controllers from Motorola (MPC555) and Texas Instruments (TMS320LF2407) are used as embedded controllers. We have implemented efficient motion planning and trajectory generation algorithms to control these robots. The capabilities of these robots were demonstrated at various DARPA PI meetings.

Our prototype robots performed satisfactorily. However, they are slow. The time for establishing and releasing suction to a large extent limit the speed. They also limit the surfaces on which these robots can be used. We have optimized the speed by installing special touch sensors and real time feedback from these sensors. Unfortunately, there are very few alternatives to suction cups if one were to design a micro- robot to climb walls. We cannot arbitrarily increase the power of the pumps and motors because they take more battery energy and increase the weight. Scientists have been studying gecko's feet and its locomotion. The dynamics are very complicated. Modeling its behavior is definitely a step in the right direction. The second and significant research focus should be on the development of an intelligent (hybrid) foot, which is capable of switching from suction cup to motion suitable on the floors. A case in a point is when a robot is deployed to climb the wall to a heating duct and travel through heating ducts. Since suction cups are used through out, the robot tends to be slow. On the other hand, if we can design an intelligent mechanism that can automatically switch from suction cups to a wheeled or belt type motion, the robot can accomplish the task lot faster. The other alternative is to design a belt consisting of tiny suction cups with valves controlling the release times. New developments in the nano-technology might help to accomplish these tasks.

This document reports research undertaken at the
U.S. Army Soldier and Biological Chemical Command,
Soldier Systems Center, Natick, MA, and has been
assigned No. NATICK/TR-04/012 in a series of reports
approved for publication.

6. References

[1] A. Nishi. "A Biped Walking Robot Capable of Moving on a Vertical Wall", Mechatronics, 2(5):543-554, 1992

[2] R. T. Pack, J. L. Christopher, and K. Kawamura, "A Rubbertuator-based Structure Climbing Inspection Robot", In Proceedings of the 1997 IEEE International Conference on Robotics and Automation, pages 1869-1874, 1997.

[3] Bahr, Li, Y. Li and M. Najafi. "Design and Suction Cup Analysis of a Wall Climbing Robot", Computers and Electrical Engineering, 22(3):193-209,1996.

[4] Yano, T., T. Suwa, M. Murakami and T. Yamamoto, "Development of a Semi Self-contained Wall Climbing Robot With Scanning Type Suction Cups", Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp900-905, 1997.

[5] J. Wilson, "First-ever Climbing Robot", in Electronics World & Wireless World, vol. 96, pp837, 1990.

[6] Briones, L. and Bustamante, P. and Serna, M.A., "Wall-climbing Robot for Inspection in Nuclear Power Plants", Proceedings of the 1994 IEEE International Conference on Robotics and Automation, vol. 2, pp1409-1414, 1994.

[7] A. Nishi, "Development of Wall-climbing Robots", Computers and Electrical Engineering, Vol. 22, Num. 2, pp123-149, 1996.

[8] Shigeo Hirose and Hiroshi Tsutsumitake, "Disk Rover: A Wall-climbing Robot Using Permanent Magnet Disks", Proceedings of the 1992 IEEE/RSJ International Conference on Itelligent Robots and Systems, pp2074-2079, 1992.

[9] G. Dangi and J. Stam and D. Aslam, "Design, Fabrication and Testing of a Smart Robotic Foot for Microrobotic System", Proceeding of the 2000 International Symposium on Robotics, 2000.

[10] Mark A. Minor, "Design and Control of Constrained Robotic Systems for Enhanced Dexterity and Mobility", Ph.D. Dissertation, Michigan State University, 2000.

[11] API-Portescap, "Miniature High Performance Motors & Peripheral Components for Motion Solution", API Motion, West Chester, PA, 1998.

[12] Maxon-Precision-Motors, "High Precision Drives and Systems Catalog", Maxon Precision Motors, Inc., Burlingame, CA, 1999.

[13] Micro-Mo-Electronics, "Miniature DC Drive Components Catalog", Faulhaber Group, Clearwater, Fl, 1997.

[14] Meng Yue and Mark Minor and Ning Xi and Ranjan Mukherjee, "Kinematics Workspace Analyses of a Miniature Walking Robot", Proceedings of the 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp 1798—1803, 1999.

[15] Dulimarta, Hans. "MPC555 Embedded Controller, Programmer's Manual", Technical Report, Electrical & Computer Engineering, Michigan State University, 2002.

[16] K. S. Fu, R.C. Gonzalez, and C.S.G Lee, "Robotics: Control, Sensing, Vision and Intelligence". Mcgraw-Hill, 1987.

[17] Texas Instruments Inc., "TMS320LF/LC240x DSP Controllers Reference Guide", 2000.

[18] Jizhong Xiao, "Development of Miniature Climbing Robots --- Modeling, Control and Motion Planning", Ph.D. dissertation, Michigan State University, 2002.

[19] R. L. Tummala, R. Mukherjee, D. Aslam, N. Xi, S. Mahadevan, J. Weng, "Reconfigurable and Adaptable Micro-robot", Proceedings of 1999 IEEE International Conference on Systems, Man and Cybernetics, pp 687—691, 1999.

[20] Jizhong Xiao, Hans Dulimarta, Zhenyu Yu, Ning Xi, R. Lal Tummala, "DSP Solution for Wall-climber Micro-robot Control Using TMS320LF2407 Chip", Proceedings of the 43rd IEEE Midwest Symposium on Circuits and Systems, 2000.

[21] Jizhong Xiao, Mark Minor, Hans Dulimarta, Ning Xi, R. Mukherjee, R. Lal Tummala, "Modeling and Control of an Under-actuated Miniature Crawler Robot", Proceedings of 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp 1546—1551, 2001.

[22] Mark Minor, Hans Dulimarta, Girish Danghi, Ranjan Mukherjee, R. Lal Tummala, Dean Aslam, "Design, Implementation and Evaluation of an Under-actuated Miniature Biped Climbing Robot", Proceeding of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp1999—2005, 2000.